High Performance Simulation and Modelling of Wireless Vehicular *ad hoc* Networks

Thomas David Hewer

A thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

of

University College London

August 2011

Declaration of Authorship

I, Thomas D. Hewer, declare that this thesis titled 'High Performance Simulation and Modelling of Wireless Vehicular *ad hoc* Networks' and the work presented in it are my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signed:

Date:

"The nation that secures control of the air will ultimately control the world."

Alexander Graham Bell

Abstract

Vehicular communications occur when two or more vehicles come into range of one another, to share data over wireless media. The applications of this communication are far-reaching, from toll collection to collision avoidance. Due to the proliferation of wireless devices and their ubiquitous nature it is now possible to operate in an *ad hoc* manner between transmitting stations. Vehicular *ad hoc* networks (VANET) are a special kind of network, that experience short link times and high levels of interference, but have the ability to present many driver information and safety solutions for the worlds roads.

Computer simulation of VANET enables rapid-prototyping and intensive exploration of systems and protocol, using highly complex and computationally expensive models and programs. Experimentation with real vehicles would be time consuming and expensive, limiting the range of study that could be achieved and therefore reducing the accuracy of analytical solutions exposed through experimentation. An extensive corpus of work on networking, traffic modelling and parallel processing algorithm has been reviewed as part of this thesis, to isolate the current state-of-the-art and examine areas for novel research.

In this thesis the value and importance of computer simulation for VANET is proposed, which explores the applications of a high-fidelity system when applied to real-world scenarios. The work is grounded on two main contributions: 1) that by using inter-vehicle communication and an advanced lane changing/merging algorithm the congestion that builds up around an obstruction on a highway can be alleviated and reduced more effectively than simple line-of-sight, even when only a proportion of the vehicles are radio equipped. 2) that the available parameter space, as large as it is, can be efficiently explored using a parallel algorithm with the NS-3 network simulation system. The large-scale simulation of VANET in highway scenarios can be used to discover universal trends and behaviours in the successful and timely delivery of data packets.

The application of VANET research has a broad scope for use in modern vehicles and the optimisation of the transmission of data is highly relevant; a large number of parameters can be tuned in a networking device, but knowing which to tune and by how much is paramount to the operation of intelligent transport systems.

Acknowledgements

First and foremost I would like to acknowledge my supervisors Prof. Peter V. Coveney and Dr. Shi Zhou.

The Centre for Computational Science is filled with researchers with broad and interesting backgrounds and I thank all the students and staff that have assisted me on my way through the PhD process. Especially I would like to thank Hugh Martin, for our frequent topical discourses, also David Wright, Simon Clifford and Owain Kenway for their scientific support and help.

My thanks to my examiners: Dr. R. Mehmood (Swansea) and Dr. Kai-Kit Wong (UCL).

Thanks also to my mother and father, Philippa and Mic, and my sister Cari.

I dedicate this thesis to Rose, my partner in crime, who has given constant support, motivation and inspiration to my life and work.

My research has been supported by British Telecom Plc., the Engineering and Physical Sciences Research Council (EPSRC) and University College London.

Contents

List of Figures

List of Tables

Abbreviations

DCF	**D**istributed **C**oordination **F**unction
DLL	**D**ata **L**ink **L**ayer of the OSI Model
DSRC	**D**irect **S**hort **R**ange **C**ommunication
ETSI	**E**uropean **T**elecommunications **S**tandards **I**nstitute
FCC	**F**ederal **C**ommunications **C**ommision
GSM	**G**lobal **S**ystem for **M**obiles
IDM	**I**ntelligent **D**river **M**odel
IP	**I**nternet **P**rotocol
ITS	**I**ntelligent **T**ransport **S**ystem
MAC	**M**edium **A**ccess **C**ontrol Layer of the OSI Model
MANET	**M**obile **A**d hoc **N**etwork
OFCOM	The **OF**fice of **COM**munications
OFDM	**O**rthogonal **F**requency **D**ivision **M**ultiplexing
OSI	**O**pen **S**ystems **I**nterconnect
PHY	**PHY**sical Layer of the OSI Model
SNR	**S**ignal to **N**oise **R**atio
SNIR	**S**ignal to **N**oise **I**ntereference **R**atio
TCP	**T**ransmission **C**ontrol **P**rotocol
UDP	**U**ser **D**atagram **P**rotocol
V2V	**V**ehicle to **V**ehicle
VANET	**V**ehicular **A**d hoc **N**etwork
WAVE	**W**ireless **A**ccess in a **V**ehicular **E**nvironment
WiFi	**Wi**reless **Fi**delity

Chapter 1

Simulation of Vehicular Networks

The evolution of wireless networks in the last decade has led to lightweight and inexpensive equipment now being fitted in many electronic devices. The majority of this network equipment uses the IEEE 802.11 [1] standard, also known as WiFi, after wireless fidelity, to ensure interoperability across different manufacturers and devices. As the size and cost of equipment continues to fall the number and types of device that will be wirelessly equipped will probably increase far in excess of the number of human users. Wireless systems in use today are ubiquitous and pervasive, the more advanced devices achieving communication with other devices without our knowledge, providing services for data and location-specific information. In a home wireless setup there is usually a wireless access point with an Internet connection, delivering data to and from attached wireless devices. The access point controls who can send and receive and coordinates 'fair use' of the medium. In the past few years, however, as devices have become more network-functional, users are increasingly expecting their devices to interoperate without centralised control. Wireless *ad hoc* networks are realised when devices fall within the transmission range of one another; on the move, links can be created and dropped in a matter of seconds. Mobile *ad hoc* networks (MANET) [2] can pair and share information with a large number of other devices over the course of their operation, leading to a sparse and intermittently connected network. However, there are challenges in MANET systems caused by the lack of centralised control and these are matched by the number of research studies and solutions put forward to address them.

A vehicular *ad hoc* network (VANET) is a MANET where all the networking devices are contained in a vehicle and, in some cases, also in road-side equipment. The ability of vehicles to share information provides the basis for a number of intelligent transport system (ITS) applications. Broy et al. [3] have shown that the software in a vehicle is as complex as any computerised mechanical system (as arises in robotics) and can be used to monitor and control most aspects of the vehicle's operation. It is not difficult to foresee how a VANET and increased complexity of vehicular technology could enable safer driving, closer monitoring of traffic systems and, in the long term, semi-automated driving. A number of ITS applications have been proposed and studied, including safety-of-life and collision avoidance systems, reaching the news in the case of [4]. Torrent-Moreno [5] has studied a number of safety applications for a VANET, where the fundamental network operation enabling safety applications is explored. The ability of vehicles on the road to know, and then communicate, their location to each other can be used in a congestion reduction application, as presented in my earlier work [6], where increased line-of-sight (LOS) to an obstacle and the transmission of that information via a VANET, can reduce congestion and in some cases prevent traffic jams, which are common to many modern road highways. Chen et al. [7] have also presented a congestion smoothing approach, using a distributed set of compute nodes, which are the members of a VANET, to compute their next movements in response to the traffic state around them.

The simulation of real-world systems has become a 'third pillar' of science alongside theory and experimentation (a video showing the concept is available in [8] and the discussions from a workshop on simulation for networks is shown by Heidemann et al. [9]). In scientific research the ability to use computer models and programs to simulate a real-world scenario allows rapid and relatively inexpensive study of complex problems. Beyond time and cost, simulation using high-performance resources can enable a view into complex systems that is not available from classic experimentation. In the area of VANET research computer simulation enables research to develop applications and models for use in real-life that can be exhaustively tested before applying to cars and drivers. In this thesis I present a rigorous study of the simulation approaches available and use complex simulations across many thousands of computer processors, in order to study network efficiency, examine possible applications and develop a methodology that can be used in a wide range of scenarios.

1.1 Contribution of this Thesis

The core contribution of this work intends to solve the problem of performing high-fidelity VANET simulations in a timely manner, and to use this simulation method to rigorously explore and examine network properties on a large scale. The research undertaken in this field previous to this thesis operated on relatively small scales or reduced computation by simplifying the simulation models. This thesis makes contributions in the following specific areas:

Lightweight, tightly-coupled, VANET simulation to increase line-of-sight and reduce congestion buildup: Many of the intended applications of VANET have aspects of semi-autonomous driving with computer control. In order to model and simulate a VANET that is capable of this the two models, networking and mobility, must be very tightly coupled during a simulation, so a closed feedback loop is available. In order to explore this loop enhancments were made to a simulation system, based on Treiber [10], that uses a tightly coupled simulation system to show how line-of-sight can be extended via VANET communication. The lane changing algorithm in the original simulation [11] has been enhanced with the knowledge gleaned from the networking side of the simulation to affect when a lane change event occurs. Through extensive testing and simulations on this system an adaptive lane-change algorithm has been developed, which produces less congestion and less velocity fluctuations (which may also lead to reduction in exhaust and an increase in petrol economy). This algorithm works when over 70% of vehicles are radio-equipped by exploring the penetration rate of equipped vehicles, which is omitted from much of the current VANET research.

An algorithm that exploits high performance computer resources that enable VANET simulations to be executed in parallel. The algorithm is used to explore a large parameter space in much less time than serial processing would require. This explorative algorithm is used to reduce the multi-dimensional parameter space to a much smaller parameter space, and a simple optimisation method is proposed: Discrete event-driven simulations, as the majority of network simulators are, present a great challenge to parallel systems. They require massive amounts of inter-process communication and suffer from causality errors. In order to explore large scale situations in a timely manner a parallel algorithm was designed in

NS-3, which runs many instances of a simulation in order to analyse different parameter changes on the network operation and efficiency. Due to the lack of intensive inter-process communication and the independence of each simulation instance, this method can explore a massive dataset in a far shorter time (hours and days rather than weeks and months) than if run in serial.

The communication density, put forward by Jiang et al. [12], combines a number of simulation parameters specific to VANET: transmission rate, transmissions range and vehicular density. The algorithm developed for NS-3 that is shown in Chapter 4 extends and proves communication density for a very large parameter set. In order to achieve computational processing in a timely fashion in a network device, the number of variables must be reduced and simplified. It was discovered that the average latency and packet success rate of a particular VANET scenario can be reduced to a single function when using communication density. This finding is of great importance to any VANET application where the network device is aware of the environment around it and can then adapt, or tune, the parameters at the physical and MAC layers, or to adapt the application parameters, in order to maintain a reasonable quality-of-service for packet reception and timeliness. Reduction of the computation required enables low-powered network devices to perform this calculation in real-time or to use a ready-made lookup table of parameters in specific scenarios.

Modern optimisation algorithms can be applied to VANET simulations in order to focus on and optimise certain metrics, like latency and packet success rate. In this area there is little previous research on VANET applications and so a simulation system that can explore a massive parameter space was designed, using the techniques implemented in NS-3 described above, which then optimises performance in the network operation. Given a particular scenario that is initialised and a high-performance computer with many processors available, it can be shown how optimisation leads to improved network performance on a global level. Furthermore, by 'tuning' the network parameters of individual stations the operational efficiency and performance of a network can be improved.

The overarching theme of this thesis is to define the bridge between complex VANET simulation methodologies and the application of those methodologies in the production of novel scientific analysis of VANET systems. In the course of this work the systems and protocols that make up a VANET system (shown in Chapter 2 on page 19) have been deeply analysed. In using NS-3, a well-used and validated simulation system, an open-source approach to the required coding is enabled, which is extensible and dynamic to the specific requirements of those using it.

1.2 Overview of this Thesis

The layout of this thesis follows a general theme that, coincidentally, also follows the chronology of my studies. In order to understand VANET simulation and the underlying technologies, from IEEE 802.11 to parallel computing paradigms, the important features are chronicled in Chapter 2 (page 19). Having read Chapter 2 anyone reading this thesis should have the essential knowledge required to understand the rest of this thesis. Network technology is filled with acronyms and some previous reading is suggested at the beginning of Chapter 2.

The main core of the thesis begins in Chapter 3 (page 52) with the examination of a Java-based coupled traffic simulation and network model. This is used to show how VANET communication can reduce congestion in a particular scenario and, importantly, test how this algorithm stands up with varying penetration of radio-equipped vehicles. This form of simulation in Chapter 3 is explicitly serial and therefore it takes time to produce meaningful and comprehensive data. In Chapter 4 (page 69) the parallel methodology used with NS-3 is presented. This parallel approach enables much larger scale simulations and a rigorous study of the available parameters to be explored more rapidly than when processed in serial. The methodology presented in Chapter 4 is used intensively in the rest of this thesis.

Chapter 5 presents the application of a parallel simulation methodology in discovering universal behaviours in VANET. The use of very large parallel processing computers (detailed in Section 2.7 on page 49) has enabled me to explore the multidimensional parameter space beyond that which has been studied before. Chapter 5 shows how essential simulation can be for VANET research, as it would be impossible to have produced the data and findings using real cars and roads. Another example of how high-performance computing simulations can produce useful applications is presented in Chapter 6, to perform optimisation of key VANET performance metrics: packet latency and packet success rate. Chapter 6 uses an optimisation algorithm to find optimum packet sizes for a given network scenario.

In Chapter 7 some final conclusions of the work is presented and some future applications of the work are proposed. Vehicles are already rolling off the production line with radio equipment installed and the number of ITS applications is increasing daily (Davies [13]), rendering this thesis increasingly pertinent to the field.

Chapter 2

Technology for Vehicular Network Simulation

The underlying technology and simulation platforms available are vital to the accurate and measurable study of vehicular networks. In this chapter the specifications, models and systems that have enabled this work are presented. This review of the technology and of the simulation platforms available establishes the current state-of-the-art of this field. There are many textbooks and documents related to networking technology including the official IEEE 802.11 standards and specifications [1]. This chapter provides enough information and explanation of the functions in networking technology, simulation systems and simulation methodologies to comprehend the core chapters of this thesis (Chapters 3-6).

2.1 Wireless Networking Technology

The advent of wireless computer communications goes back farther than most may appreciate, much farther than the computer revolution of the 1980's. Émile Baudot (1845-1903) first encoded data onto a wireless medium between 1870-1874 using 5-bit encoding known as Baudot code. Since that time wireless communication of encoded data has advanced in leaps and bounds, but the technology we predominantly use today is actually based on wired data communication protocols and functions. The IEEE 802.11 specifications, that are in use on the majority of personal computer equipment, inherit much from the wired counterpart, IEEE 802.3 [14], more commonly referred to as Ethernet. The IEEE 802.11 standards have been at the core of wireless communication since the mid-1990's, and advances are now driven by the proliferation of wireless equipment into many electronic devices.

Amongst the many sources of information related to this thesis there are three books that have been essential reading in the production of this research:

Wireless Communications [15] presents a thorough and complete course in the broad aspects of wireless data communication. Although new technology has come about since 2005 the main core of the book still remains very relevant and useful for study and research in the field.

802.11 Wireless Networks: The Definitive Guide [16] contains a vast amount of information for network researchers and is aimed at people managing live wireless networks, so includes operationally important data.

TCP/IP Sockets in C: Practical Guide for Programmers [17], which is recommended in the NS-3 documentation, contains practical insight into how the protocol stack and networking mechanisms are actually implemented at code-level. The simulation of wireless communications networks mirrors this implementation as closely as possible.

Wireless networking has become integral to modern life in many countries; with so many devices contending for the medium the complexity, and number, of mechanisms that operate the devices is increasing. Research into wireless communications touches on so many facets of the operation and usability of network-enabled devices that it makes it difficult to focus any corpus of work in the field. To that end, the main focus of this thesis is on the aspects of operation for medium control and physical transmission of the data across the medium.

The network layer of the OSI model (shown in 2.1) is primarily concerned with addressing the nodes that make up the network and the routing of information through a network. In wired and infrastructure-based wireless networks the addressing and routing of network members is essential because of the 'top-down' hierarchy of data movement and synchronisation. Routing information is important when a station wishes to transmit to a known end point and there are several methods of routing in *ad hoc* networks. Johnson *et al.*[18] show how routing can be achieved in a MANET with a dynamic

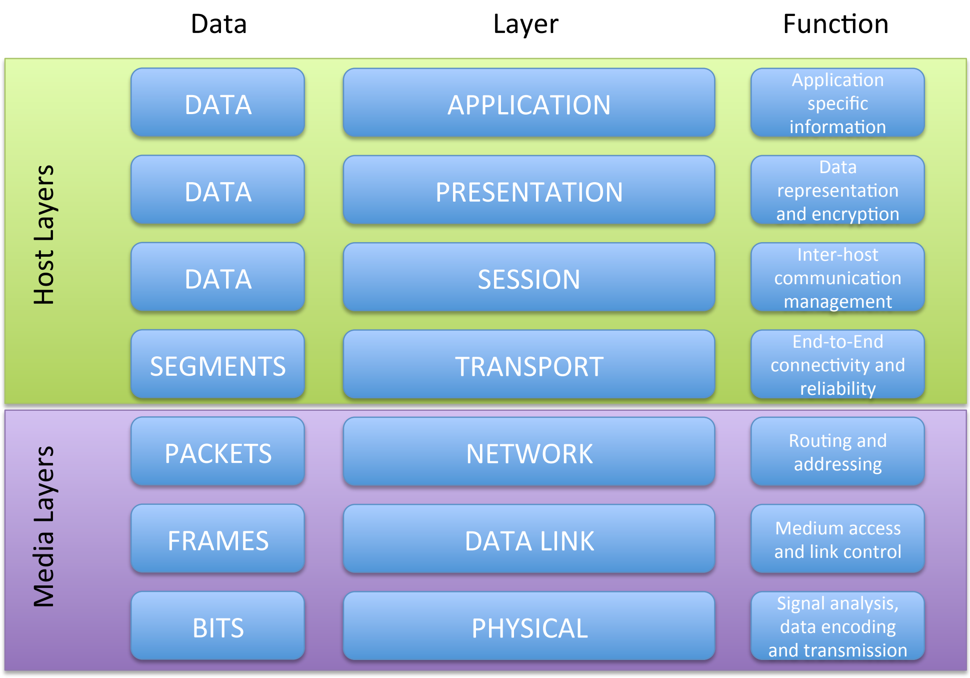


FIGURE 2.1: The open standards interconnections (OSI) model splits the communication of data down into component parts, that abstract the encapsulation of data away from the user or application. In VANET we are most interested in the MAC and PHY layers, but also in the Application layer, near the top of the model.

source approach. In this research the applications of VANET communication are primarily broadcast and single-hop in nature, so the criticality of routing and addressing is reduced, but still a factor for consideration.

2.1.1 IEEE 802.11 Wireless Communication

The IEEE 802.11 standard specifies the operation of the medium access control (MAC) and the physical (PHY) layers of the open systems interconnection (OSI) model (shown in Fig. 2.1). These layers are responsible mainly for fair sharing of the medium, robust transmission of data and flow coordination.

In 802.11 networks the medium is shared between stations within range of one another and so operates a distributed coordination function (DCF), which enables fair usage of

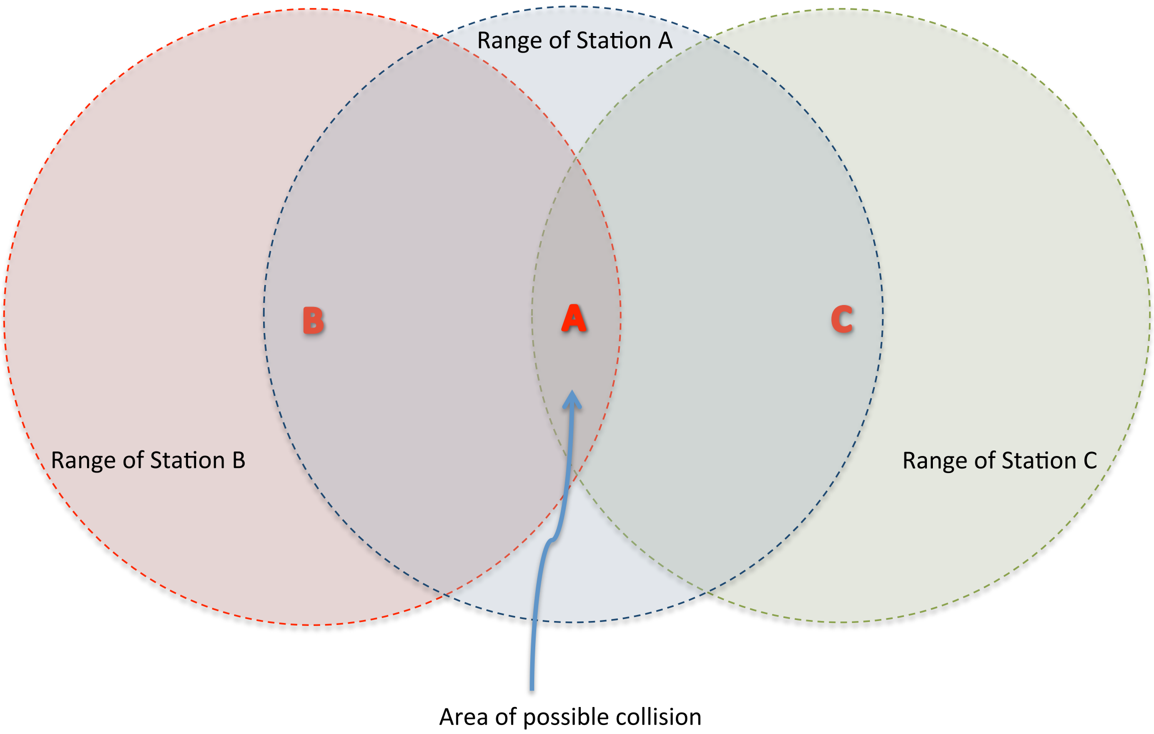


FIGURE 2.2: The hidden node effect is apparent in all wireless networks where two stations try to transmit to a neighbour when out of range of each other. The problem of hidden nodes can be mitigated by the CTS/RTS mechanism and by advanced channel assessment strategies.

the medium and tolerance for very high medium utilisation. There are several mechanisms within the DCF including a backoff function that deals with a busy medium and ensures not all stations retry at the same time. The MAC layer in 802.11 also deals with the possibility of the 'hidden node' effect, shown in Fig. 2.2. These functions are fully described in the IEEE 802.11 specifications [1].

Collisions between data on the medium is a major aspect of network efficiency and robustness, for which simulation can test and analyse well.

The physical layer of the OSI model deals with the encoding of data into binary and the transmission of this data across the medium. There are many methods for the encoding of data to binary for transmission. The main digital modulation methods are phase-shift keying (PSK) and quadrature amplitude modulation (QAM). Both methods offer a way to transmit more binary bits per symbol (a symbol is a single signal difference, in basic encoding to binary a symbol is either 0 or 1). There are variations of PSK and QAM to enable greater bit per symbol rates; the higher the bit per symbol rate is, the more sensitive to interference and lost data the transmission becomes.

A spread spectrum alternative to BSPK and QAM is direct-sequence spread spectrum (DSSS) which utilises the full spectrum available for each signal, which is encoded at a much higher frequency into 'chips' of pseudonoise. The receiver knows the encoding algorithm *a priori* and can remove the pseudonoise into the original 'chips' for decoding.

In order to further increase the bandwidth of a medium many BPSK, QAM or DSSS channels can be linked using orthogonal frequency division multiplexing (OFDM). The IEEE has an introduction to OFDM technology at [19]. OFDM enables higher datarates than traditional methods by 'linking' many channels together, but is very sensitive to interference and noise on the medium. OFDM is used in 802.11a/g/n (see Table. 2.1).

$Type$	$Freq.(GHz)$	$Modulation$	$Width(MHz)$	$Datarate(Mbit/s)$
a	5	OFDM	20	6, 9, 12, 18, 24, 36, 48, 54
b	2.4	DSSS	20	5.5, 11
g	2.4	OFDM, DSSS	20	6, 9, 12, 18, 24, 36, 48, 54
n	5	OFDM	40	15, 30, 45, 60, 90, 120, 135, 150

TABLE 2.1: The different versions of the 802.11 specification and the operating frequency, modulation, channel bandwidth and datarates available. Note: there is a 2.4GHz version of 802.11n that operates in a 20MHz wide channel, that offers half the available rates than 802.11n at 5GHz and 40MHz channel width.

The IEEE 802.11 standard was primarily designed for static and low mobility networks (i.e home/office and pedestrian), but due to the proliferation of devices the speed at which stations move is ever increasing; this leads to problems of handover between networks and delay tolerance [20]. In vehicular *ad hoc* networks these aspects are of great importance when working on application design and simulation.

2.1.2 Vehicular *ad hoc* Networking

An *ad hoc* network offers a good method to spread information outwards from an origin quickly and efficiently. It has been shown by Nekovee [21] that in *ad hoc* networks worms spread in an epidemic pattern that can be modelled. Using such modelling techniques algorithms can be developed that allow for a change to be made to the speed, position and route of a vehicle. A further advantage of *ad hoc* networking is the unlicensed use of the radio-spectrum and the recent reduction in cost for the equipment for communication. A separate strand of the research being undertaken on wireless fidelity (under IEEE standard 802.11) has been developed in the past few years specifically for vehicular *ad hoc* networks (VANETs). The 802.11p standard specifies network protocols which address the difficulties associated with vehicular networks. These difficulties involve short link times, delay tolerance and the inefficiency of wired-network paradigms, that have been inherited into the wireless standards. Some of the major vehicle manufacturers have already formed the Car to Car consortium ([22]) to share information and ensure interoperability between vehicles.

Vehicle-to-vehicle (V2V) communication is the main focus of this thesis, where vehicles may only be a member of a network for a few seconds. In a very short amount of time a great deal of information can be transmitted across the medium (see Eq. 2.1 and Eq. 2.2), but this also constrains the amount of data for more data-intensive applications.

Link times in a VANET are related to the velocities of the vehicles involved in transmission, shown by the relation:

$$T_{link} = \left(\frac{Tx_{range}}{V_{(v_1,v_2)}} \right), \tag{2.1}$$

where T_{link} is the link time, Tx_{range} is the possible transmission distance and $V_{(v_1,v_2)}$ is the difference in velocity for the sender and receiver ($V_{(v_1,v_2)} = v_1 + v_2$ when the vehicles are moving in opposite directions, and $V_{(v_1,v_2)} = |v_1 - v_2|$ when the vehicles are moving in the same direction. The amount of data that can be transmitted in this time (assuming a clear medium and no losses) depends on the channel bandwidth available

$$Th_{max} = T_{link} \times C_{bandwidth}, \tag{2.2}$$

where Th_{max} is the maximum possible throughput and $C_{bandwidth}$ is the channel bandwidth.

Vehicle-to-infrastructure (V2I) networks operating under IEEE 802.11 work in unison with some kind of road-side equipment that is mains powered and usually carries an Internet backhaul (Lochert et al. [23] offer a good method for choosing the location of road-side equipment). More commonly in V2I networks the research investigates the use of WiMAX, or mobile broadband, connections [24] because of the increased range and bandwidth availability. WiMAX is covered in the IEEE 802.16 standard [25] but is outside the scope of this thesis.

2.1.3 Propagation of Radiowaves

The propagation of radiowaves has been actively researched since the dawn of wireless communication, in order to analyse the decay and loss of signal strength over distance. The original model for how a signal will decay was put forward by Harald T. Friis in 1946, as shown in Eq. 2.3 [26] and shown in clearer detail in [27].

Friis extends the ideal free-space propagation formula ($Pr \propto 1/d^2$ where Pr is received power and d is the distance from transmitter) to incorporate the antenna gain(both transmit and receive). The Friis model, however, will only hold true with a clear line-of-sight (LOS) between transmitter and receiver, and assumes no level of scattering by atmospheric particles (which becomes very apparent in urban and highway scenarios). The spherical propagation of the wave follows $s = 4\pi d^2$ because the sphere grows in a uniform manner (where s is the surface area of the wavefront). The received power P_r at a distance d from the transmitter can be calculated as:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{2.3}$$

where: P_t is the transmitted power, G_t and G_r are the transmit and receive antenna gain, lambda (λ) is the wavelength of the transmitted signal and L is the system loss.

The two-ray model incorporates the height of the antennas into the Friis equation [28], as this will alter the total path loss caused by reflecting off the ground, and so is shown as follows:

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \tag{2.4}$$

where the values are those shown in Eq. 2.3 and h_t and h_r are the transmit and receive antenna heights.

Notice here that the wavelength is no longer a factor in the propagation, and we also drop the $4\pi^2$. This is because we are no longer calculating the omnidirectional propagation of the wave, only the path loss between two specific points.

The sensitivity of the receiving equipment is given as the lowest signal strength capable of switching from idle to clear channel assessment state 1 (CCA1), as shown in Learmonth and Holliday [29]. The sensitivity of new radio equipment is much less than even 5 years ago, but this also depends on the type of encoding the signal is transmitted with (e.g the Netgear FWG114Pv2 wireless router reports the sensitivity at http://bit.ly/fOEcos). Generally the lowest signal strength a device can successfully sense is -100dBm.

Rayleigh Fading channels (as shown in Zheng and Xiao [30]) are very important for urban areas where multi-path and shadowing of the transmitted radio wave means that the two-ray model will not present an accurate propagation loss calculation. Due to the highway nature of the simulations shown in this thesis, the two-ray ground reflection model offers the best ratio of complexity and accuracy to computational processing overhead.

2.2 Wireless Communication Specifications

The specifications and standards for wireless communication enable device manufacturers and application developers to create products which they know will work in a given situation. The development of a standard takes a long time so that all aspects of the function being specified can be developed and all stakeholders can voice their opinions. The standards available for wireless communication, such as IEEE 802.11, also enable network users and advanced operators to configure their systems to operate with the best performance available.

In 2.1.1 the IEEE 802.11 standard is introduced and in this section the standards that are specific to vehicular networking are presented. As previously mentioned, vehicular networks operate in a particular environment and setting which means the generic WiFi specifications may not provide peak efficiency and operation. The main standards and specifications are dedicated short-range communications (DSRC), IEEE 802.11p (draft) and the IEEE 1609 (draft) wireless access in vehicular environments (WAVE) specification.

2.2.1 Direct Short-Range Communications

The DSRC specifications, originally developed in the US but now in place in the EU as well, are a set of protocols and standards for operating one-way or two-way communication between vehicles in close proximity, the performance of which is studied in Khaled et al. [31]. The original intended application was for platoon driving situations, where a set of vehicles move at high-speed in unison down a road, using the inter-vehicle communication to allow very close quarters and coordinated velocity/direction changes. The main live application in use however is for electronic toll collection, where drivers entering or leaving certain road sections are automatically charged for their usage. Both the Federal Communications Commission (FCC) in the US and the European Telecommunications Standards Institute (ETSI) in the EU have allocated spectrum in the 5.9GHZ range for DSRC. DRSC is well explained in Jiang et al. [32]; in Taliwal et al. [33] the authors characterise the physical properties of OFDM technology.

The European Committee for Standardisation (CEN) specifies the DSRC physical layer in [34], the data link layer in [35] and the application layer in [36].

A notable work on DSRC was completed by Yin et al. [37] in 2003, where the authors use computer simulation of a complex road-system to measure throughput, delay and packet success. The authors found that acceptable packet latencies in VANET are in the region 70-90ms for most applications.

2.2.2 IEEE 802.11p

The IEEE 802.11p draft standard is a specification to enable communication in fast moving and dynamic environments, such as that experienced by vehicular communications. The specifications intend to provide a minimum set of standards to allow interoperability between devices using this medium environment. In vehicular networks the link times are very short and so functionality, like base station association, cannot be achieved in the time frame. Although the 802.11p specifications define more changes at the PHY than the MAC, certain MAC functionality will be specified by the standard, such as cross-layer communication (i.e. for coding method changes).

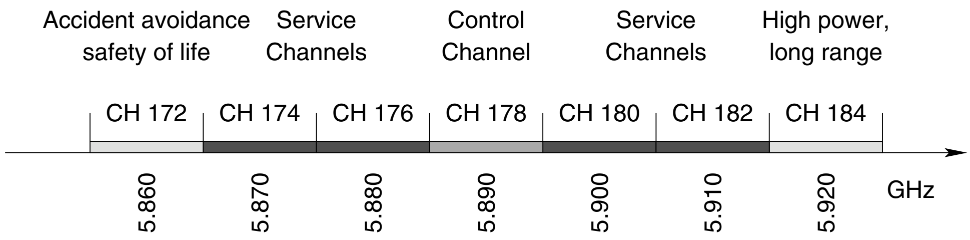


FIGURE 2.3: The seven proposed DSRC channels and their frequency bands. Each band is 10MHz wide and by using orthogonal frequency division multiplexing (OFDM) several channels can operate simultaneously.

The 802.11p standard implies a multi-channel operation, with seven 10MHz channels (there are no guard channels specified currently, so the frequency band is contiguous) at the 5850MHz-5920MHz range (usually abbreviated to the 5.9GHz range) of the spectrum. This spectrum allocation has been implemented in the US and EU and is being implemented internationally as each country reviews the spectrum allocation available. The channels are made up of a control channel (CCH), 4 service channels (SCH) a high-power and long range channel and an accident avoidance/safety of life channel. These channels are normally scanned by the equipment at predefined cycles in the order: CCH > SCH > CCH > SCH > CCH > SCH > CCH > SCH. This covers all the SCH without losing time in the CCH. Fig. 2.3 is taken from Eichler [38]).

The MAC layer in 802.11p has been determined similar to the 802.11e specification, which contains the Enhanced Distributed Channel Access (EDCA) QoS. The 802.11p channel access is split into several Access Classes (AC) that hold different settings to enable prioritisation. In much the same way as packets contend for the medium, the AC allows data to contend inside the MAC layer for access to the medium, using contention windows (CW) and back off algorithms, as in normal MAC layer contention.

The PHY layer in 802.11p uses OFDM to provide multi-channel access to the medium. This method uses many sub-carriers encoded using BPSK, QPSK or QAM (16 or 64 point), which can be changed according to the data rate required and also adapted to reduce bit error rates (BER) and improve packet reception in the network. One problem with OFDM that has been studied in Zhang et al. [39] is it's sensitivity to Doppler effects, that would be expected in highly mobile vehicular environments. There are methods to reduce the effects introduced by Doppler effects, and network performance can be sustained using different sub-carrier coding and timing.

While IEEE 802.11p is still in draft form, the latest released version of the draft allows researchers to begin testing, using computer simulation, to examine the performance of the network. Bilstrup et al. [40] have produced a detailed analysis of the 802.11p medium access control (MAC) layer and present their findings on channel access probability with different MAC settings and network load.

2.2.2.1 IEEE 1609 Draft Specifications

The IEEE 1609 wireless access in vehicular environments (WAVE) draft specification is a higher layer standard that will operate on top of IEEE 802.11p networks. The main components are for operations involving resource management, security and multi-channel operations. The working group for this standard is very active and is moving forward with the support and involvement of networking and vehicle professionals. The major stream of 1609 is in the application of security to VANET, which is caught between a need for high-speed data transfer in short periods and the requirement for securing transmitted data. Papadimitratos et al. [41] have developed a security architecture for vehicular networks, but this requires that each vehicle carry a cryptographic key which has become a problem for large-scale public-key infrastructure (PKI) systems in wired networks.

2.3 Mobility Modelling

Modelling the way in which vehicles move on roads can be simple or complex, depending on the particular road and level of simulation. When modelling vehicular networks over large areas (such as metropolitan areas) the flow of vehicles on a single road behaves as an incompressible fluid according to $Q = \rho V$, where ρ is the average density of vehicles (cars/km) and V is the average velocity on the road (km/h) [42]. At microscopic levels of simulation (across any field size) each vehicle is treated as an individual entity, which greatly increases the computational requirements of the system, but provides a more realistic and component based approach to modelling.

At a microscopic level cars often obey following rules, where a vehicle travels at a speed related to the vehicle in front and an ‘ideal’ velocity, or speed limit. The route vehicles take depends strongly on what kind of demographic the vehicle is related to. In the

TRANSIMS [43] vehicular simulation system there are options to load in census data and economic data (i.e. what type of people live in an area and how much they earn, which directly relates to their route choices and vehicle type). Microscopic simulation requires much more computational processing for a given field than at a macroscopic scale, and in highway scenarios offers little advantage.

The models available for modelling vehicular mobility vary from very simple and useful car-following models such as Treiber's intelligent driver model (IDM) [44] to TRANSIMS complex initialisation and routing features. In this thesis the Treiber IDM was favoured for mobility modelling, because of the particular scenarios being simulated and also to reduce computational processing for mobility modelling, leaving more time available for a complex network simulation. Mobility modelling can be used for driver identification in automated tolling systems, as shown in Miyajima et al. [45].

A major factor that must be appreciated in mobility modelling for VANET simulation is the massive difference in update frequency. A vehicle moving at 50mph will travel 22.352 metres per second and so the simulation system only needs to update the position of, on average, a 4 metre long car 4 times per second. In contrast a network device sending packets of 200 bytes will take approx. 0.2666ms to send one on a 6MBps channel, and therefore could send (on a clear channel with no contention or electronic delay) 3750 packets in the same time. It is vitally important to monitor this contrast in timescale and update frequency when joining simulation systems together, as discussed in Section 4.2 on page 70.

2.3.1 Treiber IDM

To accurately simulate vehicular behaviour, there are several key components: a driver model to develop how real people will drive under certain circumstances, a lane changing model to make realistic decisions on when it would be advantageous to change lane and a roadway with rules (e.g. drive on the left in the UK).

A car following algorithm will contain at least a desired velocity, a safe time separation when following other vehicles and an acceleration and braking criteria [44]. At each simulation time step the acceleration is calculated for each vehicle. The parameters of these models can be changed to emulate more aggressive and more considerate drivers, as required.

The Intelligent Driver Model (IDM) in the simulator follows the MOBIL model [11] developed by Treiber. MOBIL operates as a car-following model such that the acceleration and braking are defined by the distance from the car in front. The function of such an acceleration $\frac{dv}{dt}$ is shown in Eq. 2.5:

$$\frac{dv}{dt} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*}{s}\right)^2\right] \tag{2.5}$$

where

$$s^* = s_0 + \left(vT + \frac{v\Delta v}{2\sqrt{ab}}\right) \tag{2.6}$$

for acceleration on an open road a, velocity v, desired velocity v_0, distance s to front vehicle, desired dynamic distance to front vehicle s^*, velocity difference Δv, a safe time delay between vehicles T, a comfortable braking value b, a minimum distance between vehicles s_0 and finally an exponent δ which is adjusted in order to mimic real vehicular flow patterns.

Lane changing algorithms add a necessary level of complexity to the IDM. In order to decide whether to change lane or not, the current acceleration must be calculated for the current lane and the acceleration in the new lane (with regard to the car behind and in front in the original lane). If the acceleration in the new lane is greater than that in the current lane, there is an advantage to be gained by changing lane. Many models, including those in the original Treiber simulator, include a bias in the model for particular lanes, which simulates the real scenario of slow lanes and fast lanes.

2.3.2 Collision Avoidance Techniques

One of the major applications of ITS is in the avoidance of other vehicles on the road, especially where a collision is likely. There are several methods proposed in the literature for implementing collision avoidance systems, that all work along similar ideas. A fully automated highway, where vehicles drive automatically, have very low relative velocities and vehicular separation is very small, is technologically viable. DSRC is designed to operate in platoon driving situations. In Cho et al. [46] the authors have defined the mathematical boundaries for supporting an automatic car-following system. The work develops the idea of a minimum stopping distance to avoid collisions, given a particular velocity and heading. The concept of this system can be utilised in VANET scenarios to inform vehicles of impending approach, whether dangerous or just informational. The early warning of an approaching vehicle could be on a highway or, more likely, to operate near on and off ramps, where the vehicles need to merge with one another at high speed.

In Kato et al. [47] the authors present a cooperation and collision avoidance scheme that operates in the 5.9GHz spectrum, where vehicles transmit their location, heading and velocity (as well some other data for system control) in order to avoid collisions and maintain a high velocity. The system that is proposed uses some real-world vehicles that are set to drive at speed automatically, and the relevant information that is transmitted controls any required heading or velocity changes. The system proposed by Kato et al. [47] goes further to improve the merging of traffic driving in a platoon (i.e. driving with low relative velocity and low inter-vehicle separation).

The merging of traffic is further described in Kanaris et al. [48] where the authors have simulated and devised a method to manage the merging of a vehicle into a different lane of traffic. This is similar to the work in Chapter 3, where the vehicles are informed of an obstruction via VANET and they change lane more efficiently to avoid congestion and the obstacle. The work in Chapter 3 could be said to operate a lane merging system that recovers after the obstruction has been passed. The major difference in Chapter 3 to the work mentioned in this section, is that the advantage of changing lane increases the closer to the obstacle the vehicle becomes.

2.3.3 Simulation of Urban Mobility (SUMO)

The SUMO tool [49] is designed to be a large-scale and highly complex road transportation simulator. The models are mainly microscopic, in that each vehicle has specific characteristics and an individual routing path. The system contains many API's for extension and control and can be enhanced with census and population demographics, to create more realistic simulations. The SUMO system is advanced in that it incorporates a number of factors to create a simulation, with the ability to import GIS datasets for real-world roads. SUMO does contain a useful and complex mobility modelling system, and several networks were created for use with the NS-2 simulator.

SUMO was developed by Krajzewicz et al. [50] in order to accurately model vehicular systems and to use open-source, modular coding styles to allow enhancement and additions.

2.4 Simulation Tools for Vehicular Networking

The simulation of vehicular networks requires, as shown above, the mobility model and the network simulation, often running synchronously. Most of the simulation environments for use in vehicular environments, such as TraNS [51] which links together NS-2 (Section 2.4.2) and SUMO, were tailored from existing software and linked together to enable single-source simulation and analysis.

In the preliminary stages of this thesis the major simulations systems available were analysed. The positives and negatives of each system are shown in the following sections. Early work used the NS-2 simulator as this had a built-in model for IEEE 802.11p systems and had an intuitive scripting system. The rewrite of NS-2 into NS-3 was intensively used for the majority of this thesis, due to the performance and modelling improvements it yielded.

2.4.1 Simulation Models for Vehicular Communication

The specific implementation of the protocol stack (the lower 3 layers of the OSI model) in simulation systems varies due to coding language and author choices, but the general idea is to mimic the protocol stack as it occurs in real computer communications devices. The way that the MAC layer works, for example, in Linux mirrors closely how the MAC layer is modelled in NS-3. In NS-2 there were some problems with how the MAC and physical layer was implemented, leading to an overhaul by Chen et al. [52]. The work of Chen *et al.*was carried into NS-3 so that the MAC and physical layer models were more robust and accurate.

Network simulation software are mainly discrete, event-driven systems, where a linear list of tasks is created at initialisation that plays out according to the settings of the system. Systems such as these are very difficult to model in parallel using high performance computing resources, due to the asynchronicity of the events and the problems associated with keeping independent processing units synchronised. In Hewer et al. [53] the authors have identified the challenges facing large scale simulation in more detail, and show how parallel computing offers a solution to these challenges.

2.4.2 The Network Simulator 2 (NS-2)

The NS-2 simulator [54] was originally created as part of the Defence Advanced Research Projects Agency (DARPA) sponsored Virtual Inter-Network Testbed (VINT) project at the University of California in Berkeley (and then extended at Carnegie Mellon University for wireless networks). It has since been extended and improved with a large community of users and developers. The core kernel is written in C++ but utilises a number of Tcl scripts for the particulars of wired and wireless networks (including some details of satellite and older technologies).

The NS-2 simulation scenario scripts are written in TcL, which requires runtime compilation, that slows down the system and increases computation expense of running a complex large scale simulation. The use of TcL does simplify the creation of scenario scripts, and with deeper knowledge of the simulation system, direct C++ programs can be written.

The NS-2 system does not contain a model to simulate the IEEE 802.11p but with changes to the parameters of the simulation a single-channel operation is available, based on 802.11a, which is the basis for 802.11p. This change to parameters does not incorporate the OFDM subcarriers that 802.11p specifies, but by selecting the correct parameters a single BPSK or QAM channel can be created and set the network device to use this channel with the inter-frame spacing and bandwidth allocation, as would be seen in one of the OFDM subcarriers in a full 802.11p implementation. The lack of other subcarriers does reduce the realism of the simulation, as the signal channel is more tolerant to interference and signal attenuation.

The 802.11a MAC and physical layers in NS-2 contain full functionality, but the propagation models that have been used (two-ray ground) are only suitable for vehicular networks in a motorway situation. The use of ray-tracing and shadow propagation methods, where many routes between two nodes are calculated and the received signal strength incorporates multi-path and reflection effects, are more suitable for vehicular networks, as they more accurately represent an urban environment. The work done with NS-2 was mainly focused on motorway situations and investigating mobility patterns, so the lack of more complex propagation models did not have a negative influence on the research and results obtained.

In order to build in the applications and scenarios required for VANET research a number of code level changes were required to the NS-2 system, as well as developing a simulation script (used to set parameters and build topologies) that builds a multi-hop application with receive-and-forward message dissemination. Single-hop applications were also studied, but required less alteration to the NS-2 system. This application uses a receive-and-forward approach but retains a list of seen messages, to prevent retransmission of messages that have already been transmitted by a node.

The mobility modelling in NS-2 can be created using a number of methods. NS-2 was initially used with the independently developed BonnMotion (from the University of Bonn, created by Michael Gerharz and Christian de Waal) [55]. BonnMotion creates scenario files for NS-2 that can either contain a starting location, direction and velocity, or can be used to plot the locations of each node at a given time-step. A similar system is HWgui that was created at the University of Mannaheim by Krügler and Fūβler [56]. HWgui is only useful for simulation stretches of highway but creates good quality traces, that were statistically verified by Torrent-Moreno. Using files in this way is offline, in that the network simulation cannot change the movement of the vehicles while the simulation is running. In order to explore adaptive navigation and collision avoidance applications a fully connected model of mobility and network simulation must be in place. A full taxonomy of mobility models for vehicular simulations is covered in the work of Haerri *et al.*[57].

It is possible to feed the output of another mobility simulator, an adapted version of TrafficCom by Martin Treiber, semi-directly (still using trace files but running both simulations 'live') into NS-2 at runtime. This method allows manual adjustment of mobility model parameters at runtime, and the mobility model in TrafficCom (described in [44]) was well tested and so is less of a random-walk model, that BonnMotion is. Fig.2.4 shows the difference in computational runtime for a simulation in NS-2 with and without mobility, which shows how the overhead of mobility increases the runtime as the number of nodes increases.

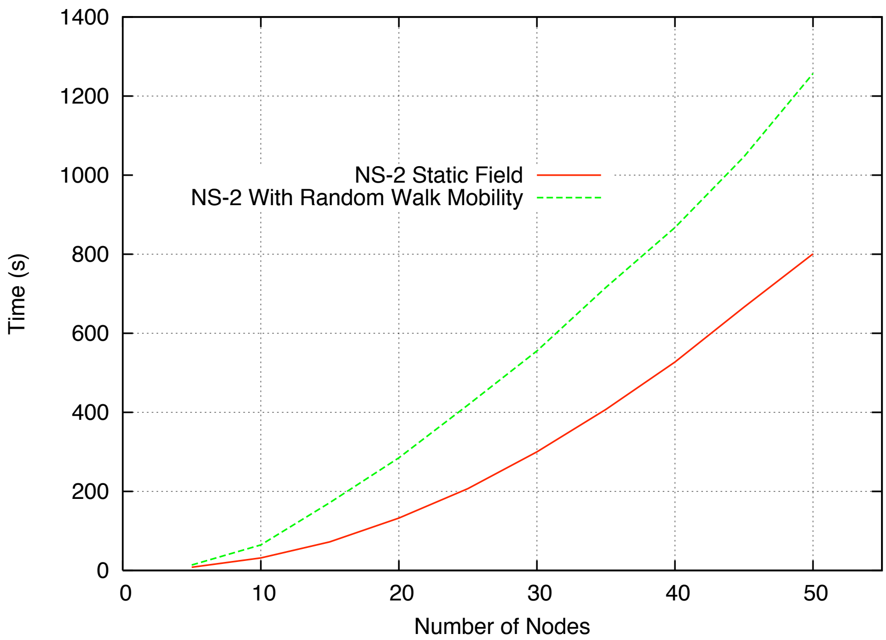


FIGURE 2.4: A comparison of runtime in NS-2 with and without vehicle mobility

The number of parameters that can be varied in NS-2 depends on which model for the MAC and PHY layers is being used, but the available parameters match closely those that can be changed in real-world devices.

The first stage of analysing NS-2 was to compare the system to other network simulators available currently. In Fig.2.5 the three main tools, NS-2, OMNET++ and Qualnet, are compared in terms of simulation runtime. Fig.2.6 shows the working memory requirements of simulations in NS-2 and OMNET++.

In order to investigate the computational expense of an NS-2 simulation, when performing a simulation of a larger number of nodes at greater density, the results from the empirically run simulation is extrapolated and then plotted. In Fig.2.7 it can be seen that the time requirements will increase with vehicular density (number of vehicles in a given field). This leads to the requirement for more than a single processor running in serial to accomplish large scale simulations in a timely manner.

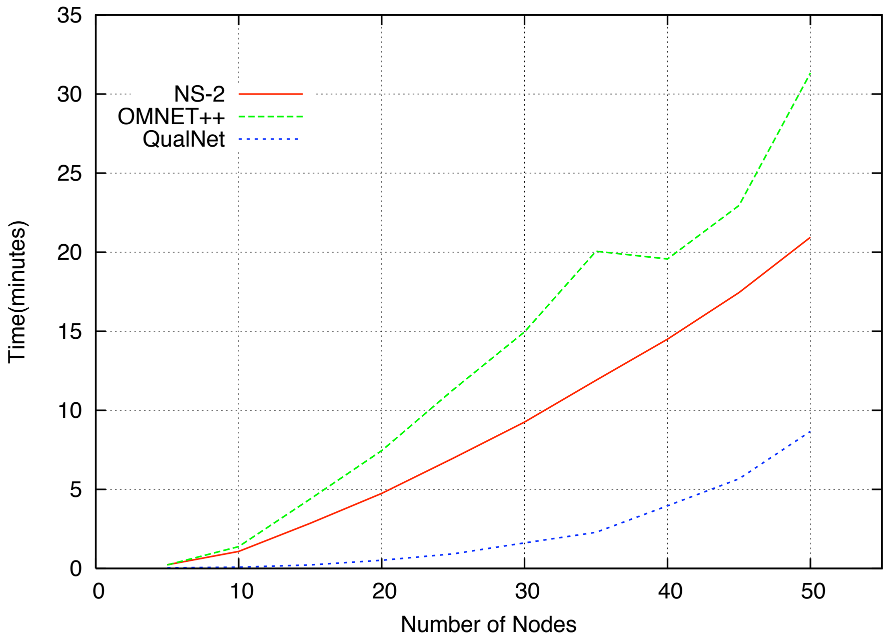


FIGURE 2.5: Simulation runtime of NS-2, OMNET++ and QualNet tested as a function of vehicular density

The work has taken advantage of NS-3, since it's release in mid-2008. NS-3 is explained further in the next section.

2.4.3 The Network Simulator 3 (NS-3)

The NS-3 simulation system is a recent, 'ground-up', rewrite of the older NS-2 simulator [58]. The simulation engine is written entirely in C++ using the latest models of telecommunications and data movement. As an event driven system, the main task is a schedule of actions to complete, in a linked and ordered list, that can be manipulated as required. The scheduler calls the sub-tasks of each simulation; in NS-3 these are C++ call-backs and functions, which run and return any required value.

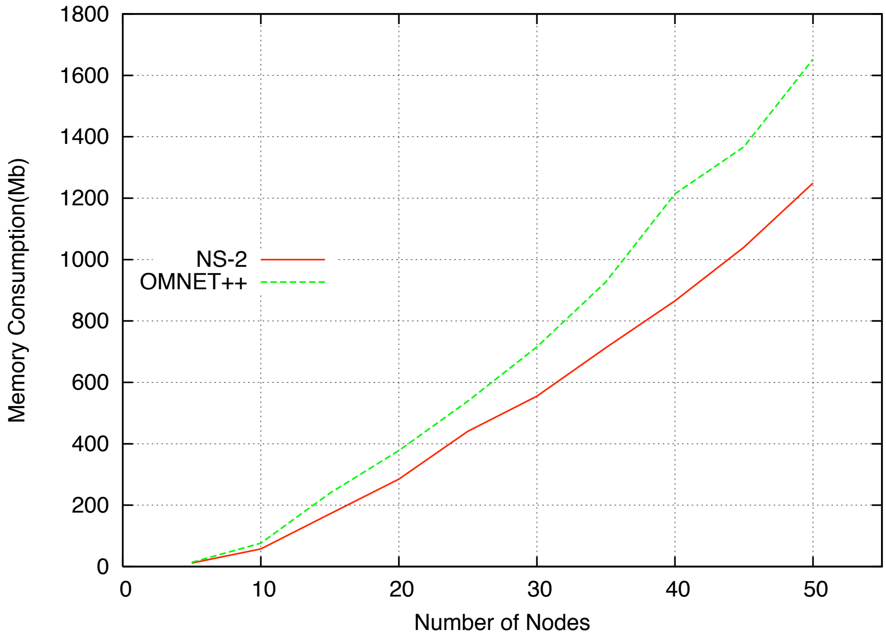


FIGURE 2.6: Working memory requirements (total) of each tool tested as a function of vehicular density

NS-3 simulations are independent programs and the methods initialise nodes, create wireless channels and install MAC and PHY models on each node, and then uses these to send and receive data and collect statistics as the simulation runs. Writing the simulation scenario is more efficient in NS-3 than NS-2, as NS-2 used an interpreted language for initialisation (The C Language (TcL)) which was bound to the underlying C/C++ functionality. The recent release of NS-3 means that there is a large amount of development activity being undertaken, and new or improved models are being released frequently (see http://www.nsnam.org/wiki/index.php/Contributed_Code). In NS-3 the simulation is written as an independent C++ program which can link into many libraries for advanced and complex functionality, most importantly the use of MPI code. The NS-3 system is fully inclusive of both wired and wireless stations, a full protocol stack (written to mirror the protocol stack in use on many *nix systems), a number of channel models, propagation and radio wave characteristics models and the ability to operate

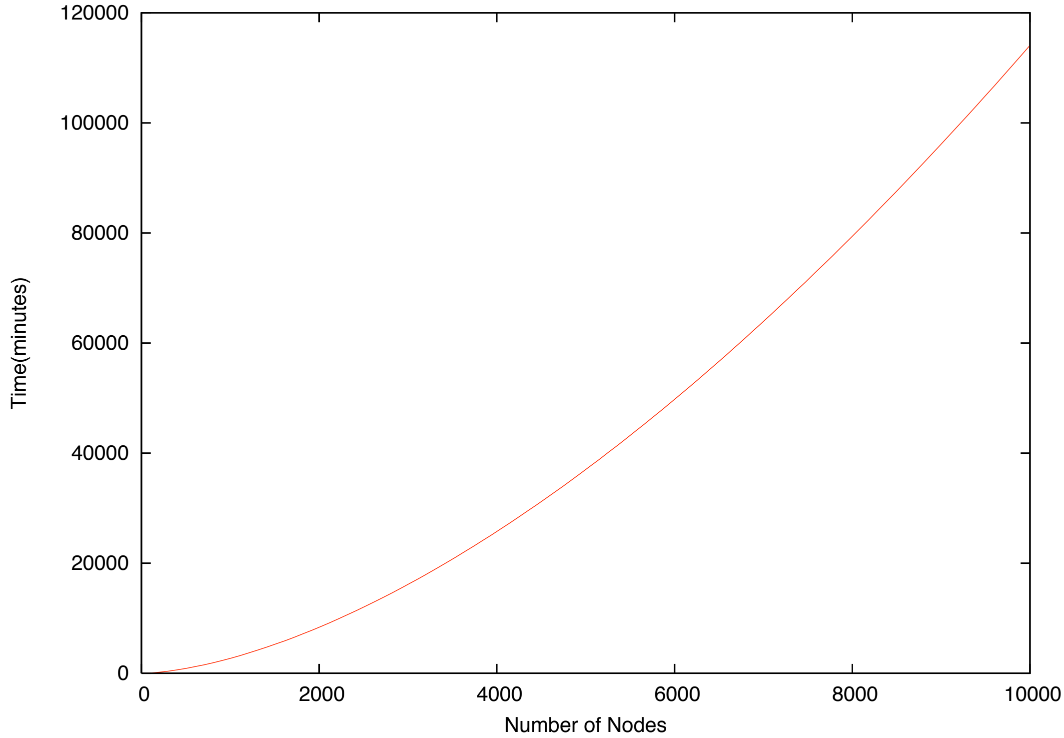


FIGURE 2.7: Extrapolated graph of simulation runtime for up to 10,000 vehicles using the NS-2 simulator

alongside a real network (the network emulation mode).

In any simulation system the models and functions provide the action and base for creating a realistic series of network events. In NS-3 the models can be broken down into several core components: common functions, core simulator functions, mobility, network devices, nodes and routing. In NS-3, as opposed to any of the other simulation systems that were analysed and tested, the code is written in an intuitive manner, as would be expected from a system developed in a single effort. The majority of other systems have been developed in a rolling fashion, that leads to difficulty keeping the code organised and therefore inhibiting users from altering and adding to the system. The 'cleanliness' of the code in NS-3 has enabled me to develop more models and advanced simulation programs.

To analyse the improvement from NS-2, the comparison simulation shown in Fig. 2.5 was recalculated including NS-3. The results show that NS-3 now runs simulations in nearly the same amount of time as QualNet, as shown in Fig. 2.8.

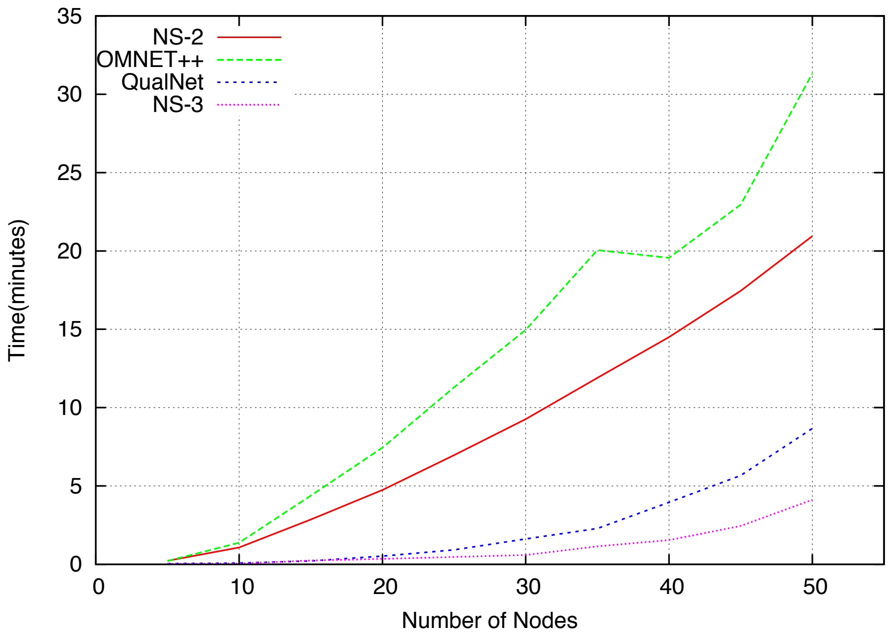


FIGURE 2.8: The comparison in runtime for a small simulation of 50 nodes as a function of time. The anomaly in the OMNET++ curve between 35-40 nodes is caused by a shift in memory allocation/management in the core system above a certain threshold.

In a vehicular environment, as specified by the 802.11p standards, the network devices use a full protocol stack, that is similar to other 802.11 standards in the application, presentation and transport layers. At the MAC and PHY layers, the system is different, and a description of the specifics of these changes is shown here.

The medium access control layer (MAC) is responsible for coordinating the access to the medium to allow multiple stations to communicate on a single channel. In 802.11p the MAC layer contains the normal distributed coordination function (DCF) that manages the slots that stations can transmit in. The slot time and inter-frame spacing is specified differently to the more commonly used 802.11a/b/g specifications. A standard 802.11a slot time is 9μs with a short inter-frame spacing (SIFS) time of 16μs, but in 802.11p this is changed to 13μs slot time and SIFS of 32μs. These larger slots and spacing times are counter-intuitive for an environment where link times are far shorter than that of fixed equipment. The reasoning behind the increased time for contention and spacing is to allow for lower bandwidth transmissions to complete successfully, and still allow fair sharing of the available bandwidth (full details of IEEE 802.11p can be found in [59]).

The original release of NS-3 did not contain an OFDM implementation, and therefore initial experiments were run using the same methodology used in NS-2. The NS-3 WiMAX module has included an OFDM channel and PHY characteristics. In IEEE 802.11p there are 64 subcarriers, 52 of which are used for actual data transmission, and 12 for control information and coordination of the medium. Using a full OFDM implementation the simulations that were performed represent a more realistic 802.11p implementation, which further validates the results that were obtained.

The EU D31 European ITS Communication Architecture specification [60] suggests that each vehicle in a collision avoidance application produce a heartbeat signal, which is single-hop and sent at 2Hz for each vehicle. In [61] the authors create a distributed fair power adjustment for vehicular networks (D-FPAV) that shows how sharing information on the state of vehicles the network can be used to coordinate the vehicles contained in a network.

2.4.4 Objective Modular Network Testbed in C++ (OMNET++)

OMNET++ is becoming heavily used in the research arena, for protocol-level development and simulation. The system offers a parallel implementation and, as with NS, is developing a number of models for simulation activities. The granular nature of the simulation language means it is ideal for simulating communications at high levels of detail. OMNET++ is well suited to educational purposes due to the graphical nature of the software [62]. The OMNET++ system is very heavily used for sensor network simulation, and Kucuk et al. [63] have developed a smart antenna model for this purpose.

2.4.5 Java in Simulation Time (JiST/SWANS)

JiST and the Scalable Wireless *ad hoc* Network Simulator (SWANS) were both designed at Cornell University. JiST was the runtime kernel, using a wholly JavaVM approach and SWANS adds the mobility and technologies specific to wireless scenarios. The user community behind JiST/SWANS is smaller than NS-2/3, but has a good amount of documentation regarding using and developing the system. In the main documentation for JiST/SWANS the designers state that it offers excellent CPU throughput in simulation time, and memory consumption [64]. The models available in JiST/SWANS are limited, as Sutaria *et al.*present [65] when they implement a new energy model. VANET simulation has been accomplished with JiST/SWANS, as shown by Schoch [66].

As stated above, the entire system is written in Java, with all libraries, configurations and mobility inputs being entered into a Java class and then compiled at simulation time. One advantage of writing in Java is that any Java web application can be entered directly into the simulation at runtime. NS-2/3 offers interfaces to attach a real-life network into the simulation runtime and applications can be run over the simulated topology, but with more difficulty than JiST/SWANS (n.b. the same difficulty applies to non-Java applications).

The mobility in JiST is limited, so SWANS provides a large amount of the necessary movements and recalculations. A simple input format is described in the documentation and this can easily be written into a JiST/SWANS configuration. One disadvantage of using the JavaVM is that any live-source data is difficult to incorporate into the simulation currently running, unless it is structured in a Java form. Both NS-2/3 and Qualnet have the advantage here as they can read files into the running simulation on the fly, and NS-2/3 is able to receive configuration changes whilst the simulation is running too.

JiST/SWANS uses a hierarchical binning system for propagation (covered further in the next chapter). In this technique the topology is split into meaningful partitions that contain a set number of nodes. This differs from NS-2/3 and Qualnet as they split the topology evenly, so some partitions contain no nodes whilst others may contain many. In the approach JiST/SWANS takes to binning the required computation to find affected nodes is less than that in NS-2/3 or Qualnet, even for mobile nodes. The available propagation models are the same as NS-2/3 (free-space and two-ray). Both NS-2/3 and Qualnet contain the option to calculate the pathloss between two nodes with differing interruptions.

2.4.6 QualNet

Qualnet is similar to NS, in that the core kernel is written in C++. Qualnet is based on the GloMoSim tool (see Nuevo [67]), that uses a different compiler to perform most simulation-time language interpretation. Parsec was designed in the computer science department of UCLA (University of California in Los Angeles) [68]. Parsec provides a parallel discrete-event driven paradigm that is highly scalable. The only major issue with Parsec is that it is supplied pre-compiled for a small number of platforms, and the operation is black-box (the inputs and outputs are known, but not the processing). This is generally because the kernel has been packaged into a commercial product, Qualnet [69]. Qualnet comes with a detailed set of documentation. Simulation setup is written into a configuration file. More models and extensions have been written for Qualnet and are provided at cost. Mobility in Qualnet is provided via a scene file sourced from the configuration file. The layout is very similar to NS-2/3, but easier to format from external sources. Qualnet may be extended but there is little support for this and the code written can only be incorporated via some key application-programmer-interfaces (APIs), unlike NS-2/3, which can be extended throughout. Qualnet offers many attractive benefits to vehicular network simulations, with a pre-built parallelised simulation-time compiler and extensible facilities (i.e. the kernel APIs). The mobility settings are very similar to NS, so the formatting of external generators would also be relatively simple. The Qualnet package does offer a good level of support and documentation, but as with most commercial packages is not open-source, providing difficulty when trying to make core changes.

2.5 VANET Communication with the NS-3 Simulator

NS-3 is the simulation system used to explore V2V wireless communication in the examined scenarios, as vehicles move in a highway topology defined by the IDM model. The NS-3 code is produced as a replacement to NS-2 with a ‘ground-up’ rewrite of the code. The physical layer model in NS-2 suffered from some problems in implementation that caused the accuracy and realism of the model to be doubtful. The upgrade to NS-3 took advantage of the work of Chen et al. [52] to rework the model and mitigate the problems.

NS-3 is a discrete event-driven system, where a central list of tasks to be completed is processed sequentially in the order in which each event should happen. The discrete nature of the dynamics means that non-trivial parallelisation of the system is difficult, and could lead to high levels of processor intercommunications that might increase, rather than reduce, simulation runtime, unless handled very carefully. The parallel algorithm developed (Section 4.5 on page 73) instead runs many serial instances of a master simulation concurrently with minor changes to various parameters to be studied. The following sections explain the deeper functions of NS-3 models for the data link layer (DLL).

2.5.1 MAC and Physical Layer Implementation

In simulating wireless fidelity networks, the majority of simulations use the IEEE 802.11 protocols [1] , as these offers the best simulation of the MAC layer functionality. The IEEE 802.11 MAC layer uses a distributed co-ordination function (DCF), which has been simulated in [70], to ensure efficient communication on the medium, and implements controls to reduce collisions. More recently the IEEE 802.11p standard has been tested in [37] specifically for inter-vehicular communications. This allows the foundation of underlying strengths in the 802.11 suite to be enhanced for vehicular networks.

The MAC layer model in NS-3 contains *ad hoc* wifi functionality and settings. The protocol stack is installed in NS-3 using helpers, where core MAC functions are extended by specific sub-models (in this case the ns3::AdhocWifiMac class). This MAC model contains cross-layer communication for medium sensing, a backoff timer as part of the distributed coordinated function (DCF) and trace sources to analyse data passing through the MAC layer on each vehicle.

The configuration of the MAC layer is in line with recent IEEE 802.11p and the US dedicated short-range communications (DSRC) specifications. In the simulations the slot, DIFS, SIFS, ACK and RTS/CTS timeouts (all terms defined in [1]) are set to accurately represent a real network device in a vehicular environment.

The DLL contains the functions relating to medium sensing, medium access control (MAC), the distributed coordination function (DCF) and queuing packets for transmission. The full standard for IEEE 802.11 wireless networks is published in IEEE [1] and under this specification there exist a number of sub-standards for different functions and environments. The most widely known are IEEE 802.11a/b/g/n, as these are the most commonly used in home wireless systems. In this section the specifics of the IEEE 802.11p draft standard [71] is presented, as implemented in the NS-3 simulation system.

The MAC layer of the protocol stack is designed to coordinate the sharing of the medium, so that many transceivers may use the same spectrum. In 802.11 networks the medium is wireless and therefore more complex to manage access control, compared to IEEE 802.3 Ethernet connections. The 802.11p MAC layer is identical to the 802.11e [72] MAC functions as it uses Enhanced Distributed Channel Access (EDCA) and Quality of Service (QoS) extensions. The fact that 802.11p will work in the 5.9GHz spectrum does not impact the MAC layer. The MAC parameters that are set in this model are shown in Table 2.2, with a comparison with the default settings for 802.11b/g. In most network devices, these parameters can be modified to achieve more suitability to the environment, within limits of operation. The DCF and medium sensing in the NS-3 model of 802.11p uses the non-QoS station functions (in NS-3 this is the NqosWifiMac under the AdhocWifiMac model). This creates a DCF with high medium utilisation due to the removal of the QoS overheads.

Parameter	*p*	*b/g*	*a*	*Explanation*
CTS Timeout(ms)	0.0756	0.0756	0.0756	limit for clear-to-send response
Ack Timeout(ms)	0.0756	0.0756	0.0756	limit for packet ack
SIFS (ms)	0.0320	0.0160	0.0160	short interframe spacing time
DIFS (ms)	0.0250	0.0300	0.0250	PCF interframe spacing time
Slot Time(ms)	0.0130	0.0090	0.0090	base for propagation time
Max MSDU Size (bytes)	2304	2304	2304	max size before fragmentation

Table 2.2: The parameters for operation of the DLL in 802.11p compared with those for 802.11b/g and 802.11a. The changes may seem negligible in user terms but are critical and highly sensitive in the transmission equipment.

As Table 2.2 shows, the differences in the DLL parameters are very small. The major difference is the increase of the slot time value, which increases the theoretical time required for transmission and allows for a lower bandwidth connection (DSRC suggests a maximum of 6Mbps). With this increase slot time the SIFS value is also increased to double the period of 802.11a/b/g. This extension to slot time and SIFS increases the available and expected time for stations to access the medium and transmit data. This information is passed down to the physical layer to assist with accurate transmission.

Parameter	*Value*	*Explanation*
EDT(dBm)	-105	Energy Detection threshold
CCA1T(dBm)	-85	Threshold to switch to CCA1
TxPower(dBm)	13/17/20/24	Transmission power

TABLE 2.3: The parameters for operation of the physical layer. EDT and CCA1 can be set according to the particular design of the network equipment, down to a lower threshold of sensitivity. Modern equipment can operate with lower EDT and CCA1 values, mainly due to noise reduction in the device circuitry.

Table 2.3 shows the main parameters in the physical layer. The most important are the energy detection threshold (EDT), which is the value of power sensed on the medium above which a signal could be detected (i.e. above background noise), the CCA1 threshold, above which the physical device can switch to CCA1 (clear-channel assessment mode 1, explained in Fig. 2.9) and the power of transmission within the medium.

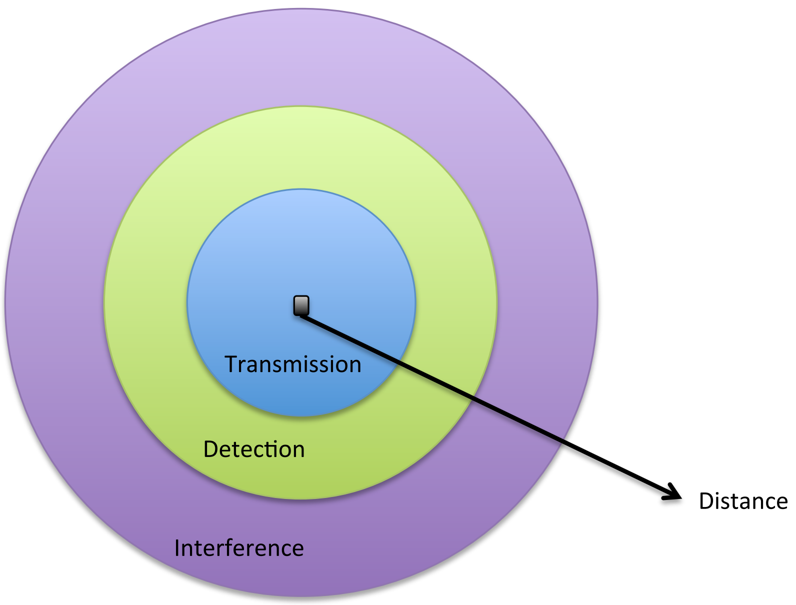


FIGURE 2.9: Figure showing the three level boundary when calculating radiowave propagation. The 'Transmission' range is that where a received signal can switch the idle physical layer to CCA1. 'Detection' refers to the receiver being able to identify the signal as data, but due to it's low strength cannot successfully switch to CCA1 and the 'Interference' range which covers signal strength below Detection and down to a strength where they no longer cause noticeable interference at the receiving node.

The physical layer in NS-3 uses the Yet Another Network Simulator (YANS) model [73]. This model describes the IEEE 802.11a specification and functionality and contains calculations for Bit Error Rate (BER), that is the percentage of received bits that are in error during transmission; expressed as a number referenced to a power of ten.

The YANS model includes Forward Error Correction functionality, where bits are subjected to inner convolutional coding before transmission, such that bits received in error can be detected and reassembly attempted at the receiver.

The PHY model can operate in one of four states, representing the different activities undertaken at this layer of the OSI model. The states are:

- TX: transmitting data onto the medium;

- SYNC: PHY has detected data on the medium and is awaiting completion of the transmission;

- CCA_BUSY: Clear Channel Assessment (CCA) cannot detect data on the medium, but the energy measured on the medium is above the Energy Detection Threshold (EDT);

- IDLE: PHY is not in TX or SYNC states.

Packet reception is not completed if PHY is in either the TX or SYNC states, or if the received power of the data is below the EDT. If either of these situations occur the packet is dropped. If PHY is available for reception (i.e. in the IDLE state) and the packet reception energy is above the EDT the PHY changes state to SYNC until reception is complete. Once the last bit of the packet has been received the probability of reception is calculated using the Signal to Noise Interference Ratio (SNIR) function given in detail by Lacage and Henderson [73] and shown in the NS-3 source code [74].

This probability rate defines whether or not the packet is received by a node within range, and simulates the effects of a real network through interference, multi-path propagation and system error. The interference helper model accounts for bit error and packet errors and calculates whether a packet is received, assuming it is received with enough power to switch the PHY state to SYNC, by performing a probabilistic calculation on the bits received which, according to a threshold, leads to the packet error rate (between 0-1) for that packet.

$$SNIR(k,t) = \frac{S_k(t)}{N_i(k,t) + N_f}, \tag{2.7}$$

where $SNIR(k,t)$ is the signal-to-noise interference ratio for packet k at time t, $S(k,t)$ is the energy of the received signal, N_f represents the noise floor and $N_i(k,t)$ is the interference noise (the sum of all energy of other signals including, mainly, thermal noise):

$$N_i(k,t) = \sum_{m \neq k} S(m,t). \tag{2.8}$$

The channels available for wireless stations are a basic WiFi channel, with free-space and two-ray ground propagation models, and the YANS WiFi channel, which, as well as extending the basic WiFi channel, allows for more complex propagation and delay models, to accurately represent the channel effects that would occur in a real network.

The YANS WiFi channel operates by creating the channel characteristics and calling the propagation and delay models, then applying a Send() function, to allow data packets to be sent through the channel. Once the channel has been installed on the nodes, they are able to send data but, without a MAC or PHY model, this data would simply be point-to-point and not as accurate as a wireless channel.

2.6 Parallel Programming with MPI

When processing code on parallel processing machines, the code must be decomposed such that smaller units of the whole can be processed independently. The Message Passing Library (MPI) provides the functionality to separate code and allows inter-process communication in a lightweight, therefore fast, way. The MPI compilers are 'wrapped' around the normal GNU C++ compilers, such that the non-parallel parts of the code are compiled as normal. This leads to a very portable system that can be used on most parallel-enabled computers.

In the work of Miguel et al. [75] the authors explore the parallel methodology available for event-driven simulations of communications networks. This work is important in this thesis and includes many aspects and methodologies that can be transplanted to the NS-3 system. Although the authors work was undertaken in 1996, and technology has advanced greatly since then in terms of HPC size and capacity, the ideas they put forward, such as decomposing the operating units of a network device for operation on separate processing units, are still useful today.

The core commands in MPI are listed in Table 2.4:

MPI_Init	Initialise the MPI library in the code
MPI_Comm_size	Finds the number of processors that can be used
MPI_Comm_rank	An identifier for the processor that this code is running on
MPI_Send	send a number of data to another rank
MPI_Recv	receive data from another rank

Table 2.4: The core MPI commands that make up a basic MPI program.

There are many more commands in the MPI library, and many different implementations. Often a high performance computer will include various versions of MPI, as each has specific functionality that code is developed to take advantage of.

2.7 Implementing NS-3 on High Performance Systems

The NS-3 simulation system was written for serial processing, and therefore adapting it to run on the high-performance machines listed in this section has been very challenging. A serial processing code will call functions under different headers and in different code sections, that will return values to the simulation program. In this way, if the processing units are separated to cover different functions or different areas of the simulation, this code must be updated so that it can find the calling code.

NS-3 has been implemented with parallel capability through MPI, but this currently only covers point-to-point and CSMA (carrier-sensing multiple access) which represent wired communications [76]. Although not extended to cover wireless communication, the addition of MPI headers to the core simulation system opened an interface into this functionality.

In the following sections a brief overview of the HPC systems used in this thesis is shown.

2.7.1 Legion Dell/Intel Cluster

Legion is a 710 node (at the time of writing) cluster machine at UCL. Each node has 8 Intel processors and between 2-4Gb of RAM with an Infiniband interconnect to a Lustre filesystem. The version of MPI in use on Legion is the OpenMPI [77] branch, that uses common and generic names for the MPI commands in use. In order to install and run NS-3 on Legion the latest Python shared libraries, bzip2 compression libraries were required and updates made to a number of NS-3 headers, so that they included the MPI headers. Legion is made up of Dell 1U servers, and as such the compilers and libraries available are similar to those found on a default installation of Linux. This leads to NS-3 running well and fast on this system.

Legion is the main system used for the research to date, the resource allocation is performed in a fair-share system, where queues are controlled by time and number of cores required, which can lead to long waiting times for jobs to be initialised. At the time of writing over 4,000 simulations have been performed on Legion.

2.7.2 Mavrino Sun Grid Engine Cluster

The Mavrino system is a 144-core Sun Grid Engine machine, that is owned and operated by the Centre for Computational Science. There are two main machines types in Mavrino: 10 4-core Intel processors with 120Gb local storage and 12 8-core machines with Intel processors and 16Gb of RAM. The interconnect on Mavrino is Gigabit Ethernet and there are NFS mounts available to offload data. In the initial stages of this research Mavrino was used for testing and development, but it has also been well-used for the main research and to perform 'production' simulations. Installing NS-3 on Mavrino required the same changes as made with Legion, and the compilers used (again, OpenMPI) mean that NS-3 runs well on this system.

The back end storage on Mavrino is limited, and so data must be compressed and removed to alternative storage as the simulations finish, but this only requires minor additions to the submission script. As Mavrino is only used by a small research group, the queue times are often low, but the lower number of cores available mean that there is a limit to how much processing can be performed. Several hundred jobs were run on Mavrino, and the results were compared with Legion output, to ensure there are no artifactual errors introduced by a different system.

Chapter 3

Congestion Reduction Using Vehicular *ad hoc* Networks

The congestion of vehicles on highways creates a number of negative impacts on modern society: the loss of time, the cost of petrol and the problems of greenhouse gasses. The number of vehicles in operation is always increasing (apart from recent financial effects [78]) to the point where there are now more than 30 million cars on the roads in the United Kingdom. When one lane of a highway is obstructed by an incident the weight of traffic is forced into a smaller area, leading to congestion that can travel backwards from the obstruction many miles.

The algorithm developed in this chapter takes a coupled approach to simulation, with the networking and mobility modelling interlinked at runtime. Using VANET to increase the LOS to an incident ahead and using this information to alter a lane-change decision can directly influence the build up of congestion and flow of vehicles past an obstruction. The scenarios used to study this algorithm operates on a dual-carriageway environment with no junctions, with an obstacle or danger that is present at some point in the field. The simulations use both vehicular flow modelling and message propagation to advise the vehicles of the obstacle at a greater distance than line-of-sight provides. The tight coupling of communications and mobility models at runtime reduces the complexity of the simulations (as shown in [79]). This method of simulation can also be useful for intelligent transport systems, where simulations run in parallel can sweep the parameter-space for a desired outcome. The impact of applications such as this is to reduce the time spent in congested vehicular flow but also to increase the safety interval and reduce collisions. Many possible applications of VANET technology are designed to improve safety-of-life while driving and in this simulation it is shown that the application of novel algorithms can realise this goal.

The use of satellite communication, such as those linked to global positioning services (GPS), offer global communication but require expensive equipment, large antennae (for two-ray transmission) and, due to the large distances the signal must travel, have a high latency. Cellular telephone networks offer a lower latency over large distances but are still slow when communicating with nearby vehicles, due to a centralised approach, and require cellular contracts to use the network. Most of the applications suitable for VANET require high-speed communication disseminated from source, which is difficult to achieve using either cellular or satellite communications. VANET is specifically suitable to the scenario discussed in this chapter.

The simulation tool used is adapted from the dynamic vehicular flow simulator by Treiber *et al.*[44]. This tool uses a simple model of a two-lane roadway, but contains an advanced driver model and lane changing algorithm, MOBIL [11]. The model and algorithms that run it are shown in 2.3.1 (page 30). Telecommunications aspects have been added to the original simulator for V2V communication and this has been developed further with enhanced algorithms and functionality by Nekovee and Bogason [42].

3.1 Concept

When a driver approaches a blocked lane or accident on a highway they can only see the vehicles in front slowing down. This means that as congestion builds while others drivers try to avoid the blocked lane, the distance of informed drivers reduces significantly. This experiment shows that by using vehicle-to-vehicle communication in an *ad hoc* way, the distance of informed drivers from the actual obstruction can be increased. The reception and processing of this information by drivers can allow them to make alternative lane change decisions, and in a more complex system to use alternative routes. The algorithms shown here and tested through simulation could be used to update satellite navigation systems or to design highly adaptive road traffic information (i.e. signage or public radio).

3.2 Simplified MAC Layer Protocol

The MAC layer in the simulator operates using an adapted version of IEEE 802.11 which removes the inter-frame spacing (IFS) model, enabling equal priority for all network traffic. Due to implementation, and the need for simplicity in the model, this implementation of 802.11 does not suspend the back-off timer when the medium is busy during that frame, as the full version of 802.11 does.

The network back-off when the medium is busy X operates as shown in Eq. 3.1 [42]:

$$X \in 2^n \times [B_{min}, B_{max}] \tag{3.1}$$

where n is the number of times it has previously had to back off in succession. B_{min} and B_{max} are the minimum and maximum possible back off time, respectively. B_{min} is often set at 0. The medium is defined as busy if any car within the transmitters interference range, R_i, is currently broadcasting.

Every car within the transmission range, represented by R_c (which is usually twice as small as the interference range), will receive the message with probability λ.

3.3 Simulation Setup and Scenario

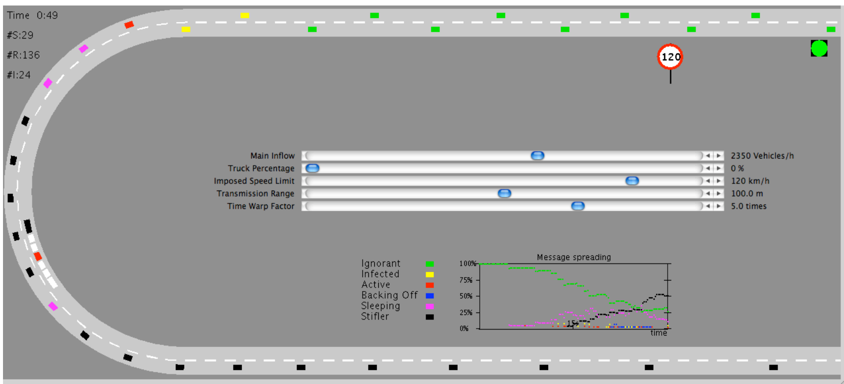


FIGURE 3.1: The simulator interface with the cars entering in the top left and exiting at the bottom left, with the obstacle shown in the lower quadrant of the curve to the left. The sliders in the middle of the screen control vehicle inflow, speed limit and transmission range.

In order to deliver usable and realistic results, the simulated system must be an accurate representation of a real highway. The metrics gathered with the results obtained are shown in Sections 3.7-3.10. The system used for the study is shown in Fig. 3.1. The road length is approx. $1400m$ and has two lanes with vehicles moving in the same direction and an obstacle at $720m$, in the left-hand lane. Vehicles originate in the top-right of the road and travel around to the departure point in the bottom-right. The flow of vehicles entering the road can be varied from $0 - 4000$ vehicles per hour, split across the two lanes. The desired speed (v_0 in the Treiber IDM (Section 2.3.1)) can be varied between $20 - 140km/h$, which will directly affect the vehicular density as greater velocity will require greater separation, according to the IDM.

The transmission of data between vehicles is controlled by them operating in five distinct states; (i) ignorant, where the vehicle has received no data and is not transmitting; (ii) informed, when the vehicle receives data from the network; (iii) transmitting, to re-send received data through the network; (iv) backing off, when the vehicle wishes to transmit but the medium is busy; (v) sleeping, when the vehicle has forwarded data but is waiting to make sure all neighbours have received that data; (vi) stifling, when the vehicle has performed all the reception/transmission required and therefore is no longer active. The movement through these states, from ignorant to stifler, can be seen as a loop. The propagation methods described in Section 3.6 control how the vehicle progresses through the states in different ways. The transmission range of the vehicles is also set globally for the simulation, between 0-200m (it is possible to extend the transmission range of vehicular networks beyond 200m but only by using very high transmission power and assuming little interference). The vehicle that is causing the obstacle (surrounded by several car-lengths of safety barriers) constantly transmits without moving to the sleeping or stifler states, so that any vehicle coming within range will receive the data, and then begin the process of receive-and-forward.

3.4 Variable Speed Limit

The initial algorithm explored reduces the value of the desired velocity (v_0 in the Treiber IDM 2.3.1) by a fixed amount when the vehicle has received the warning message. This achieves an overall slowdown in the network which can reduce the time delay between free-flow and gridlock (where $v = 0$) at the obstacle. This particular algorithm change has a transient effect on the network, such that the system will still become congested over time. The idea for this came from the London Orbital (M25) variable speed limit which operates on parts of the motorway.

The value that v_0 is reduced by is of great importance. Initial tests showed that reducing the desired velocity too much (i.e. a reduction of over 10m/s) when infected, the effect on the network was to cause congestion further back in the system, such that gridlock (i.e all vehicles in the field of simulation are static) occurs much sooner. By reducing the value of v_0 by 2.7m/s the network slowed well and the time delay between free-flow and gridlock was increased without causing congestion further back in the network.

An important algorithmic change is the return to the normal value of v_0 once the obstacle has been passed geographically, otherwise the recovery from the obstacle will take a greater amount of time. A proportional change in the desired velocity as the obstacle approached was examined, but this provided little observable effect at great distances and a highly-negative effect closer to the obstacle, as cars were congesting more smoothly but to a greater extent.

The results of this algorithm change were both transient and often detrimental to the overall velocity of vehicles in the system, so the change was dropped from the final algorithm described in Section 3.5. The reason for this negative effect is thought to be related to the finite size of the field being simulated. In later simulations (see Chapter 5 on page 80) a much larger field is explored, where it is expected that the effect of the algorithm becomes more pronounced as the distance from the obstacle increases.

3.5 Enhanced Lane Changing Algorithm

The original lane change model operates by determining an acceleration advantage (toward a goal velocity) to be gained by changing lane and then testing if a gain threshold is reached by the advantage. Early incarnations of the changes to the algorithm worked to forcibly increase the advantage if a message had been received and the vehicle was in the lane with the obstacle. This has some positive effects, but can cause problems when the message propagates a large distance back through the system. In the case of the message propagating beyond the reasonable extent of the need to change lane, this approach causes unnecessary congestion in the adjacent lane to the obstacle resulting in total congestion in a short time.

The lane changing algorithm proposed by Treiber can be referred to as a lane merging operation, described more in Section 2.3.2 on page 31, due to the merging of traffic to avoid the obstacle and the way in which the traffic recovers after the obstacle has been passed.

At the start of the simulation several static variables are applied to the model. A changing threshold is applied that indicates the increased acceleration the lane change will yield; this is set by default at 0.3ms^{-2} for cars in the field. The other value is the 'politeness factor' which reduces the overall calculated advantage and which simulates the care drivers take when changing lane (i.e. the model may say it is advantageous to change lane, but the driver may be too polite or hesitant to do so).

The basic algorithm operates by calculating a value of advantage (A), the disadvantage this causes to other (B) and then calculates whether a function of these values reaches a changing threshold. If the threshold is reached the vehicle changes lane.

$$A = a_n - a_o + B, \tag{3.2}$$

where a_n is the acceleration in the new lane and a_o is the acceleration in the old lane. B refers to a weighting to keep the vehicle in the slow lane, as operates in reality,

$$B = a_{b(o)} - a_{b(n)}, \tag{3.3}$$

where $a_{b(o)}$ is the acceleration of the car behind in the old lane if the vehicles changes lane and $a_{b(n)}$ is the acceleration of the car behind in the new lane if one were to change lane. These values are then entered into the following equation to return true or false to changing lane:

$$(A - p)B > T, \tag{3.4}$$

where p is the politeness factor and T is the changing threshold. The form of Eq. 3.4 is multiplicative so the values of A and B have a significant impact on each other.

Equations Eq. 3.5 - 3.7 show the enhanced design I developed; these enhancements are only used if the vehicle has been infected with a message. If a vehicle is in the ignorant state (explained in Section 3.3) it will continue to use the Treiber algorithm in Eq. 3.4.

The initial change made was to add a value to A in Eq. 3.5 as such:

$$((A+V)-p)B > T. \tag{3.5}$$

This is very much a brute force approach and as such does not truly represent real driving in a highway, where the value of V would increase as the vehicle approaches the obstacle and drop to zero after the obstacle has been passed. This proportional addition to A is shown in Eq. 3.6 and Eq. 3.7:

$$\tau = \frac{P_o}{(P_m - P_o)} \tag{3.6}$$

$$((A+\tau)-p)B > T \tag{3.7}$$

In this adaption of the original algorithm the value τ is calculated as a function of the location of the obstacle and the vehicle's distance to it. This value is capped at a maximum (currently 20) to prevent unrealistic behaviour (i.e. cutting in with zero safety headway), which means the effect is noticeable but quite subtle, when compared to the brute-force method in Eq. 3.5 . This means that as the vehicle approaches the obstacle the incentive to change lane becomes greater, reducing the appearance of congestion at the obstacle in the same lane.

3.6 Transmission Method Experiments

In this experiment the available transmission methods are tested with the new algorithm, to see how they affect the overall congestion in the system. Fig. 3.2 shows a simulation that was run until congestion is present at the origin of the field (i.e. position = 0). The simulation is stopped when the congestion reaches the origin as after this point the algorithm is not affecting the vehicular flow. Fig.3.2 shows the vehicular throughput of the system over time, where flow profile is cars exiting divided by cars arriving in the field. This indicates how vehicles are flowing through the system, where an increase of the gradient represents free flow and a decrease represents congestion.

The models shown in Fig. 3.2 are; (i) no propagation, where no radio signals are being sent; (ii) simple flooding, where the message is rebroadcast just once; (iii) edge detection, where vehicles probabilistically retransmit when they sense they are near other vehicles and back-off when there are no other vehicles nearby; (iv) distance detection, where a bias is applied to the probability of re-broadcast according to the physical distance from the source and (v) a mixture of edge and distance detection. These models are all available in the simulation, but the mixed (edge detection and distance detection) offers reliable and bandwidth-efficient delivery, as found in [42].

As can be seen from Fig. 3.2 the edge detection method alone offers little improvement over no propagation, and the simple flooding and distance detection methods offer a good initial advantage (0-200 seconds) but then suffer very fast congestion build-up. The mixture of edge and distance detection algorithm, with the changes to the lane change model offers excellent results keeping near free flow until ca. 420 seconds, when the network then slows and starts to congest, but this takes longer (ca. 250 seconds from the first slowdown) than the other algorithms.

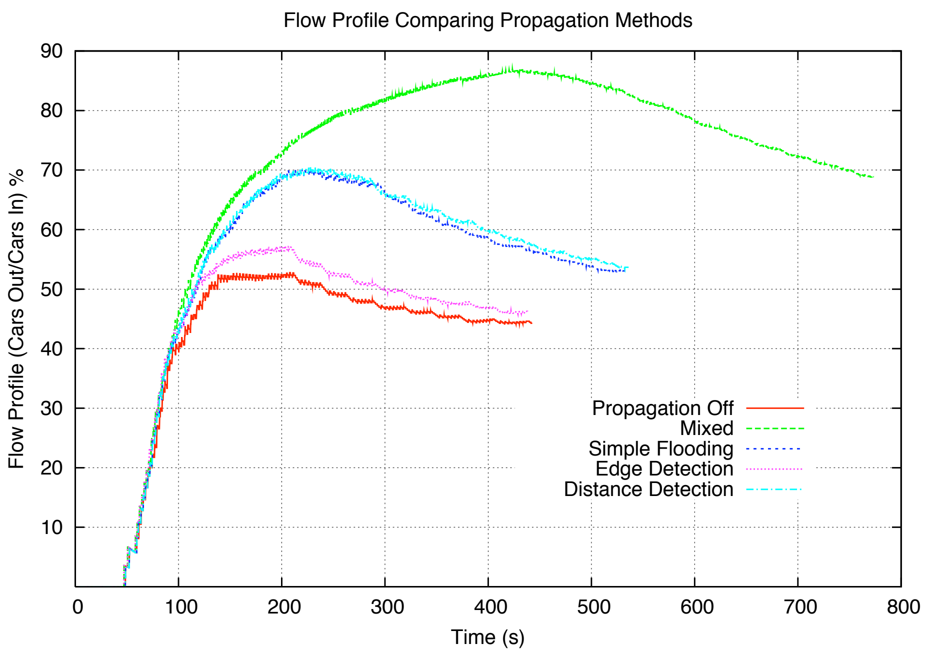


Figure 3.2: Comparison of message propagation methodologies in an obstructed highway showing the vehicular throughput for each methodology

The addition of this propagation method and the changed algorithm prevent several congestion-causing situations to occur. The main situation avoided is when vehicles are unable to change lane to avoid the obstacle and begin to slow down, but then do change lane causing the cars behind to slow. This cause has been seen to initiate the build-up of congestion, but by earlier warning of the obstacle the cars can change lane at a higher velocity. Another behaviour of the system is that when the congestion is initiated, the cars will fill up behind the obstacle, unable to change lane. With the adapted algorithm the extra incentive to change lane means that the opposite lane fills first and so vehicles can still move, increasing the time before the whole system becomes congested.

3.7 Performance of Lane Change Algorithm for Varying Velocities

These simulations test the effectiveness of the algorithm as vehicle velocity changes, to see if the algorithms are suited to an urban (slow) or highway (fast) environment. When ignorant (no messages have been received), the cars will still attempt to change lane to avoid the obstacle, but only as part of the original lane changing algorithm, and so congestion builds up in a short amount of time, for most simulations. Below a certain network load the road will never become congested, so the vehicular load of the experiment was varied for each experiment. The vehicular load on the road was also set low enough so that the algorithm could affect the flow of cars as, at high loads, this would not be possible. To this end there is, in any system, a critical value of vehicular load after which no action can prevent or reduce congestion.

Some early simulations with low vehicular loads showed that it is sometimes more efficient to be ignorant of the obstacle, and this must be taken into account, as in this case the best course of action is to drive normally, using the normal algorithm.

The following charts show results from a sample of the simulations that were run to test this theory. By varying all the parameters available it was found that certain velocities, transmission ranges and vehicular load had different levels of effectiveness to the overall congestion in the system. In the main the results showed that the new algorithm always produced a positive effect. In order to produce valid data several simulations were run with the same parameters and the output was averaged, to show results for a typical case.

Figures 3.3 and 3.4 show the number of cars exiting the field in a simulation run as an aggregate over time. Both show an advantage for infected cars using the advanced lane changing algorithm, but the advantage is greater at higher velocities, where the vehicles have more distance between them for the same vehicular load, meaning they can more easily change lane.

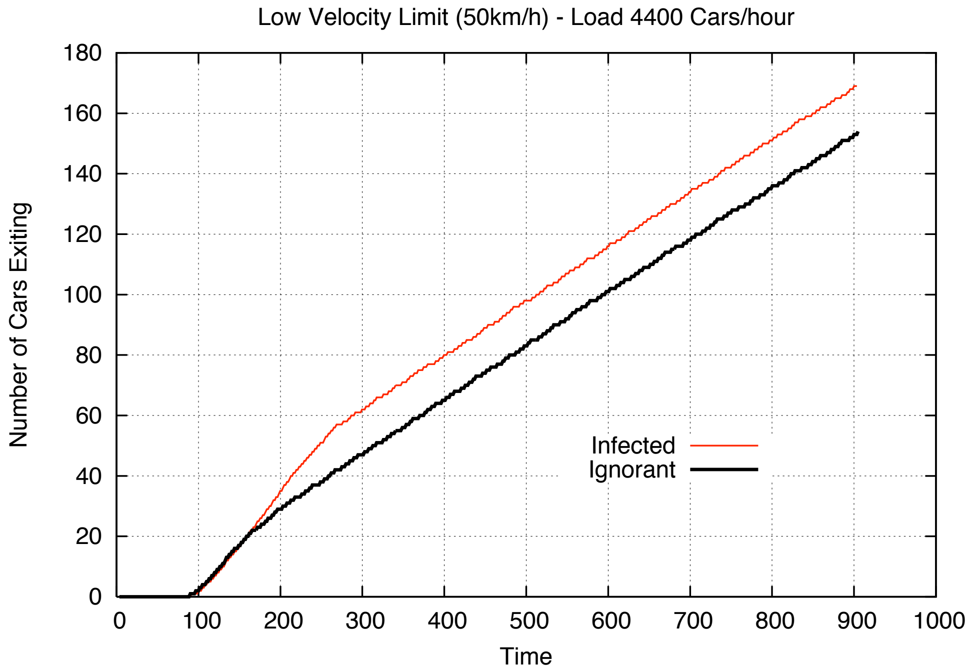


FIGURE 3.3: Comparison of exit aggregate for simulations where radio is off (vehicles are ignorant) and where messages are being sent about the obstruction (vehicles are infected with the message) at urban velocities (below 50km/h)

It was found, and Fig. 3.3 and Fig. 3.4 corroborate this, that the advanced algorithm increased the time before congestion began to build up and then once congested, the infected cars still moved through the system more efficiently. To monitor the development of congestion in the system and how the flow is affected by the changed algorithm the scenario is simulated for 15 minutes (simulation time).

The results show interesting behaviour, beyond the reduction of congestion in the system. By analysing the first 5-7 minutes of simulation time, the results show that the development of congestion is also slower once vehicles do start to slow down. This is because of the algorithm moving vehicles into the opposite lane to the obstacle, reducing the load on the lane with the obstacle and therefore reducing the number of stopped vehicles behind the obstacle which, when changing lane, cause a dramatic slowdown in the new lane. This reduction in stop-and-go vehicular formation is also seen elsewhere in the field when the cars are infected with the warning message and switch to the adapted algorithm.

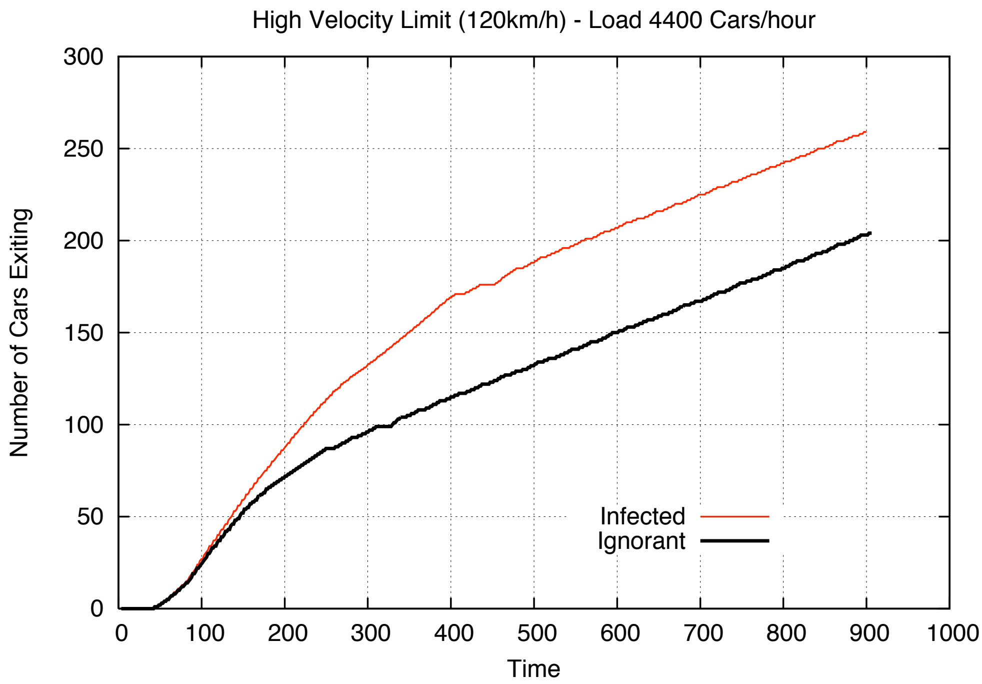


FIGURE 3.4: Comparison of exit aggregate for simulations where radio is off (vehicles are ignorant) and where messages are being sent about the obstruction (vehicles are infected with the message) for motorway velocities

3.8 Position of Lane Change Event as a Factor in Congestion

A factor that affects the build-up of congestion in the system is related to the location of the lane change. The following results show where the lane change occurs with no communication and then using the enhanced lane change algorithm with communication active. The simulation settings were set at 2200cars/hour load, speed limit of 120km/h and a transmission range of 100m.

During the simulation the message propagates backwards towards position 0 and, with the advanced algorithm, the location of the lane-change also reduces. When the system starts to slow and vehicular density increases, the lane-change moves right back, causing a slower build-up of congestion and a greater amount of free vehicular flow, as shown by Fig 3.5. This does place greater load on the opposite lane to the obstacle, but the reduction of stop-and-go behaviour negates this. In Fig. 3.5 the rapid reduction in position of most lane changes between 395-405 seconds and again at 475-500 seconds represents a period when the propagation of the message is continuous, and the periods of little change (of lane change position) are due to reduced propagation of the warning message. The initial peak of lane change position between 0-35 seconds represents the initialisation of the system, that cars can change lane very close to the obstacle due to the road being less loaded. The data to produce this result came from a single typical simulation, with parameters set as per the previous paragraph.

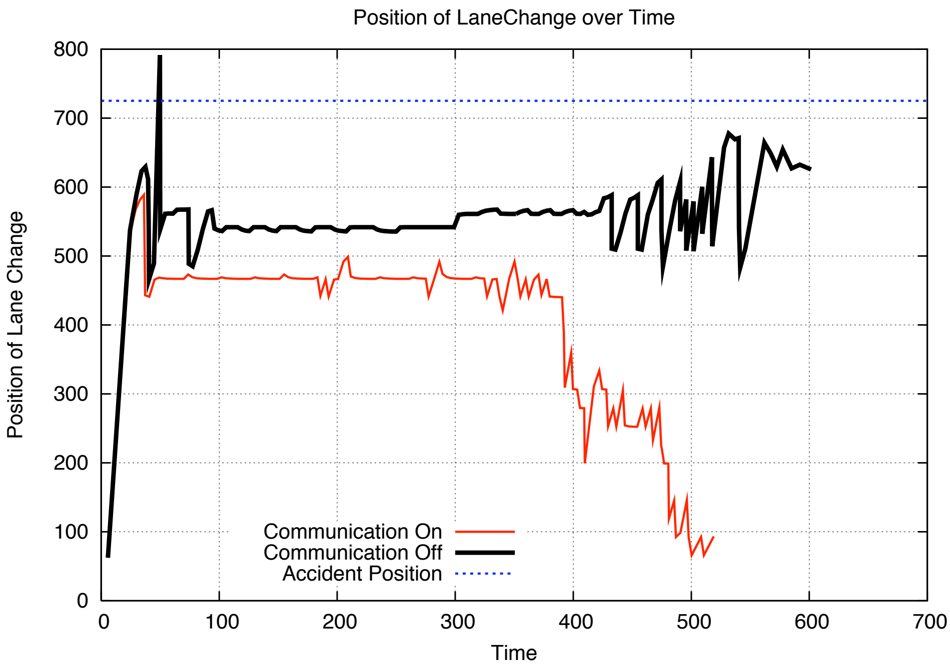


FIGURE 3.5: Chart showing location on the field of lane changes with and without communication

3.9 Average Velocity of Vehicles Through the Simulated System

The average velocity through a system is of great importance. If a higher average velocity can be achieved the number of vehicles passing through the area of congestion will be higher than if there is much slowing of vehicles. The following two figures show the average velocity calculated for intervals of 10 metres on the x axis and an interval of 30 seconds across the y axis. Each point represents the average velocity at that time/position interval. In both figures there is a noticeable slowdown as the vehicles pass the obstacle. This can be accounted for by the IDM attempting to retain a minimum safe distance between vehicles.

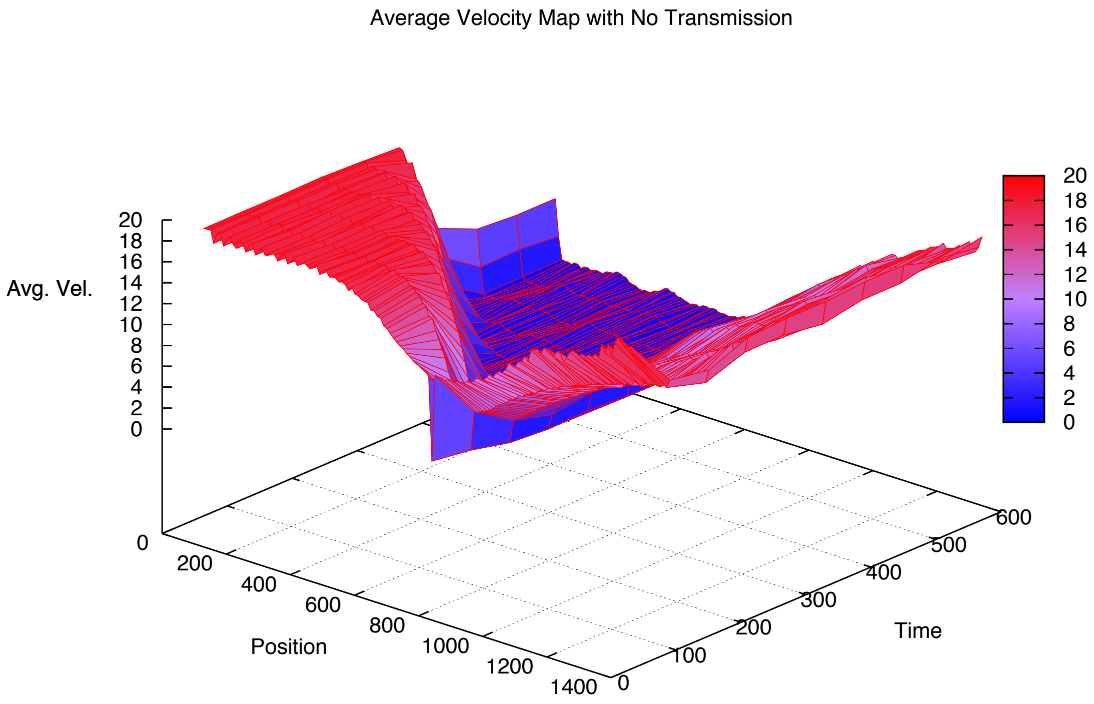


FIGURE 3.6: Average velocities across the simulated highway without transmission

Fig. 3.6 shows that after an initial even velocity through the field, congestion begins to build up at the position of the obstacle between 100-200 seconds, which causes a slowdown further back to position 0. By time 330 seconds the congestion has reached position 0 and the average velocity falls from 20-25ms^{-1} to 0-5ms^{-1}.

As can be seen in Fig. 3.7, there is a uniform average velocity before and after the obstacle during the whole period of the simulation (10 minutes). This reinforces the results from the other simulations and proves there is a better flow of vehicles through the network, as well as a reduction in the build-up of congestion, when effective transmission of the road condition occurs.

A point of note for both figures is that there is a spike in velocities (between time 0-10 and position 1200-1400). This is the period when the first cars are leaving the field. As they have no cars in front they can accelerate up to the full speed limit unhindered. To

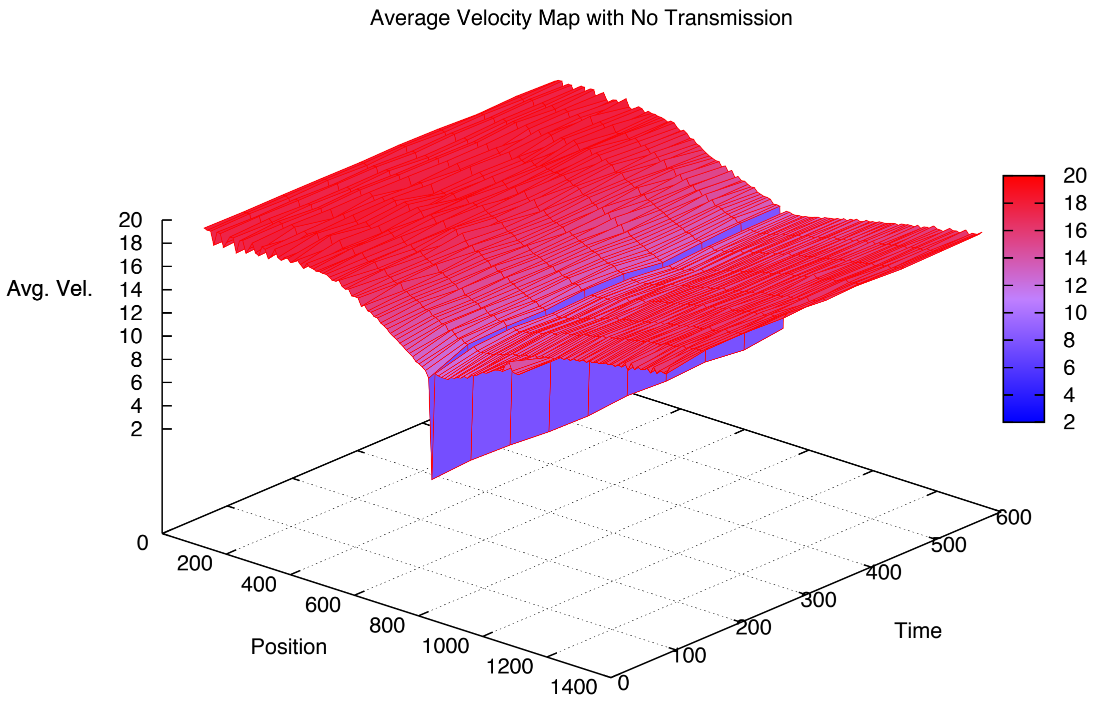
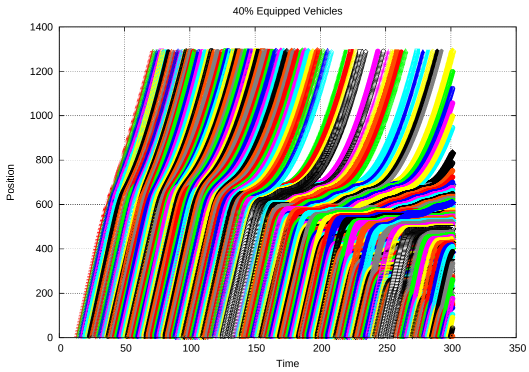


Figure 3.7: Average velocities across the simulated highway with active transmission

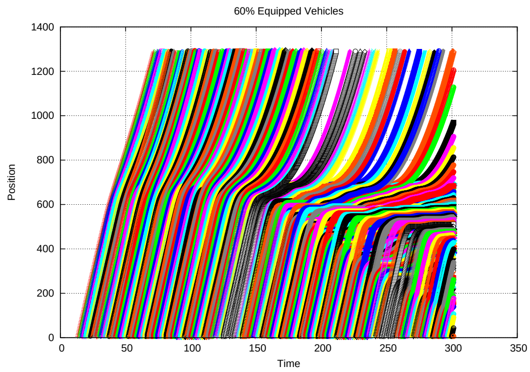
remove this artefact future simulations will have a 'warm-up' period, with low vehicular load, that initialises the field.

3.10 Analysing the Effect of Penetration of Radio-Equipped Vehicles

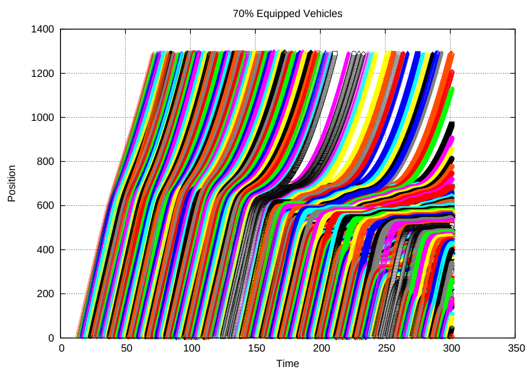
In all of the simulation studies and results presented in Section 3.7 - 3.8, it was assumed that 100% of the vehicles in the simulated field are radio-equipped and therefore able to receive and re-transmit (where necessary) the data about the upcoming incident or congestion. This is a somewhat idealised view of how deep uptake of radio-equipped vehicles will be. More likely, as vehicle manufacturers and radio equipment providers begin releasing the products to operate and support a VANET, the ratio of equipped vehicles will increase. Due to the critical safety-of-life applications of VANET, it is important to study the efficiency and resilience of simulated applications, to study the point at which they become usable and reliable. This simulation study involved the same system as in previous experiments, but here the proportion of equipped vehicles is increased from 10% to 90%. The plots in Fig. 3.8 shows the flow trajectories of vehicles in the system.



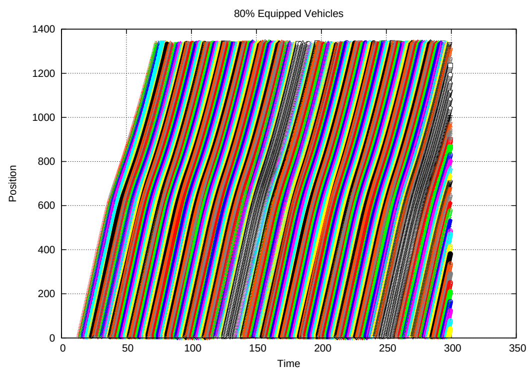
(a) 40% Equipped



(b) 60% Equipped



(c) 70% Equipped



(d) 80% Equipped

FIGURE 3.8: Vehicle trajectories through the system for 40% (top left) equipped vehicles, where little change can be seen from 0% equipped vehicles, 60% (top right) where some improvement is shown but congestion still builds up, 70% (bottom left) where the improvement is noticeable and 80% (bottom right) where the trajectory is only slightly affected by the obstacle. Each coloured trajectory represents a single vehicle as it enters and leaves the simulated highway.

As can be seen from Fig. 3.8 at low penetration rates (10-40%) the vehicles reach the obstruction and begin to congest, but between 60% and 70% there is a noticeable improvement, or smoothing, of the vehicles velocity through the field of study. The figures show that, above 70%, where there is still some congestion build-up around the obstruction, the propagation of information to vehicles further away, and the change to the lane changing algorithm this triggers, allows for more steady driving through the field of study, and less congestion.

3.11 Conclusions

This chapter has presented a simulation of a specific road condition, that of an accident blocking a lane on a dual carriageway. The simulation uses the coupled approach to mobility modelling and network simulation in a single process. The models and algorithms that represent the vehicular flow and network traffic have been implemented according to well-known standard models. The results show that adapting the algorithms when information about the accident is received via wireless transmission can reduce the build-up of congestion and increase the flow of vehicles passing the obstacle, both in terms of quantity of cars and average velocity.

The various algorithms simulated achieved overall improvement in the majority of cases. In some simulations the improvement was not only in the prevention of congestion overall, but by also keeping a consistent average velocity through the network, which helps to reduce the effects of stop-and-go vehicular flow and smooth possible congestion 'waves' that emanate from the source backwards. The driver model and lane changing algorithms come from well validated sources, and so the adaptions made are highly realistic and can show the effect of even simple changes (as in the brute force addition of a value to A). In order to fully test these algorithms numerous tests were run with a wide variety of parameters, to test for any transient or artifactual effects. From these repetitive runs it can be established that most effects were long-lasting and that where those were transient, this was only due to a vehicular load on the road where congestion could not be prevented (as corroborated by control tests).

In the common situation where the obstacle is temporary (i.e. a vehicle malfunction), any reduction in congestion build-up allows for the obstacle to be removed, before the velocity of all vehicles behind the obstacle drops to zero. In more complex roadways the advanced warning could also lead to a change of route, so drivers can avoid the section of road where the obstacle exists.

The real-world application of this algorithm is limited to enhancing driver information and may not operate well when a driver ignores the recommendation for lane change. One application of this algorithm is for an advanced driver information system, where the details of the obstruction and suggested actions are displayed in-car. The modelling of penetration rates can also be used to mimic the impact of the obedience of human drivers, when there is 100% penetration. In the future, when automated driving systems are in place and more relied on, such an algorithm could form a part of the plethora of functions and processes used to drive a vehicle under computer control.

Büscher *et al.*[80] discuss the challenges of congestion reduction systems, from a social perspective. The authors work motivates the need for accurate human-behaviour modelling in vehicular network simulation and what impact the social aspects of mobility provide. The simulation system is designed to perform these studies using accurate mobility modelling, but for a real implementation the social aspects of the driver and the context in which they are flowing becomes essential.

The size of field simulated here is between the microscopic and macroscopic scale of simulation, which is achieved seamlessly by the use of a coupled model of simulation. The tool used to perform the simulations was lightweight and so new algorithms and protocols could be easily implemented. In order to run more complex networks the system would require a more complex simulation engine for vehicle flow. With this increase in complexity the field size will increase and therefore a more powerful network simulator is required.

Chapter 4

Parallel Parameter Exploration for Vehicular Network Simulation

Using high performance computing paradigms and resources to run many simulations in parallel the results can be obtained in much less time than would be achieved through serial computation. A typical NS-3 simulation takes $60s$ to run, and over $1,000$ parameters sets are explored. Using traditional serial computing this would take 16 hours of continuous runtime, but the runtime can be reduced to ca. $60s$ to complete all simulations (including the overhead from initialising the HPC resources). The methodology for running and analysing output incorporates node aggregation (running many short simulations in serial, whilst in parallel with larger, long running simulations) and automated feedback of parameters. The technique of feeding parameters back into a simulation after runtime can lead to a convergence, where a parameter 'set' provides a network with a reception probability that is adequate to avoid collisions. In this work many such simulations are carried out and the results and analysis of the data is provided, so it may be applied to large-scale and highly complex networks.

4.1 Concept

The simulation of vehicular networks using computers creates a rapid-prototyping environment that can then be examined in a number of scenarios. Given that the simulation tools available (in this case NS-3) are capable of simulating complex vehicular ad hoc networks, the use of high performance computer resources can be tailored to run faster and many more simulation instances than by using serial processing computers. In order to fully utilise the available computer resource the algorithms that were developed are efficient and streamlined to reduce lag from unnecessary overhead.

The algorithm presented in this chapter underpins the work on universal behaviours in VANET (Chapter 5) and on the optimisation algorithm developed in Chapter 6.

4.2 Coupling

In order to accurately model a vehicular network, the vehicles must be mobile, such that changes in density and separation of the vehicles affects the communications efficiency and success. The topology should be created with a real road network in mind, with intersections and lanes where vehicles can reside.

The IDM in the simulator follows the MOBIL model [11] which was developed by M. Treiber (described in 2.3.1 on Page 30). A major factor in the coupling of mobility to telecommunications in a simulation is that the two models operate at different update intervals. A vehicle can only move a certain distance in 1 second (e.g. 13.41m at 30Mph or approximately 3 car lengths), whereas many thousands of events can occur in a protocol stack and medium. The synchronisation of the models was achieved by merging the functions of each model as shown in Fig. 4.1 and Fig. 4.2.

4.3 NS3 Models and Configuration

The NS-3 simulator [81] is presented in 2.4.3. A major advantage of NS3 is the ability to create standalone simulation 'programs' that only need to link to the NS-3 library after compilation. This lightweight and portable approach has enabled the simulations to use over 256 cores (not the full allocation but the most that can be achieved while competing for access with other users, as per the queuing policies) on the UCL Legion cluster machine, and offers the ability to scale to many thousands of cores.

4.3.1 Application

The scenario explored for this work is a stretch of highway, 2km long with 2 lanes. The vehicles are positioned according to the IDM and the ends of the highway are

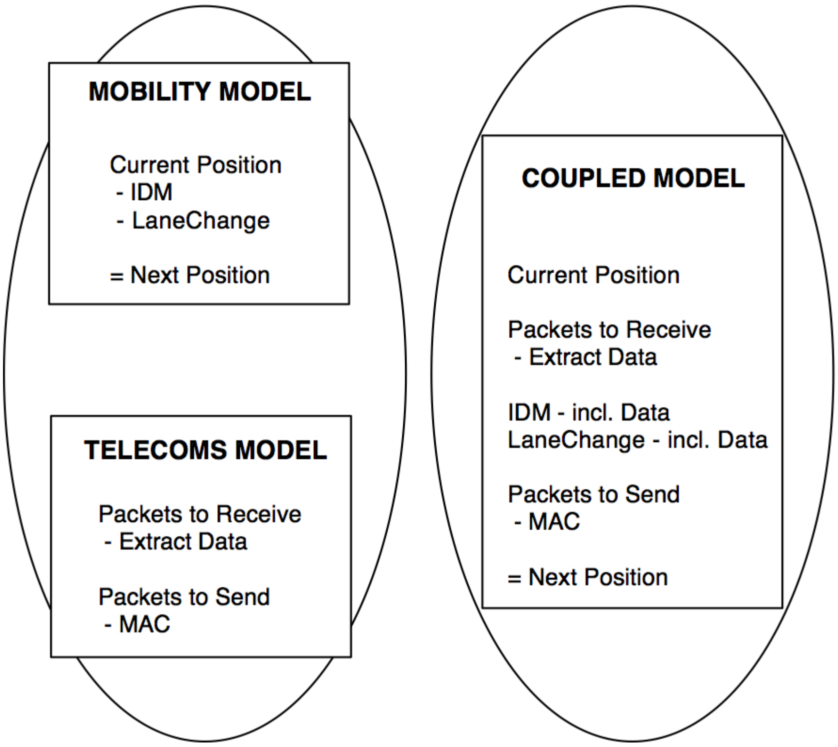


FIGURE 4.1: Merging of mobility and communications models

'wrapped-around' to reduce boundary conditions (i.e. where a vehicle has no neighbours in one direction). The vehicles are set to transmit informational messages at a given interval that is defined by a Poisson distribution across a value (in this case 1-100 packets per second). In the EU ITS architecture specifications [60], these 'heartbeat' messages are sent at a rate of 2 packets per vehicle, per second (ppvps), but in this work the exploration went above and below this value, in order to investigate how the network is affected. The packet is made up of a unique identifier, a location (from an on-board GPS receiver), bearing and velocity, but can be adapted to contain more or less information. Packet fragmentation should not occur in most applications as the packet size will remain below the 2348b fragmentation threshold set in the NS-3 MAC model (as defined in [1]).

The Poisson distribution gives the greatest level of realism for a simulated highway environment, rather than suggesting all vehicles will transmit simultaneously; the vehicles are unsynchronised and cannot coordinate sending times, which could reduce collisions on the medium.

The messages are varied in size and are broadcasted over the medium in a single-hop manner, so no vehicle must re-transmit a received packet. This transmission method

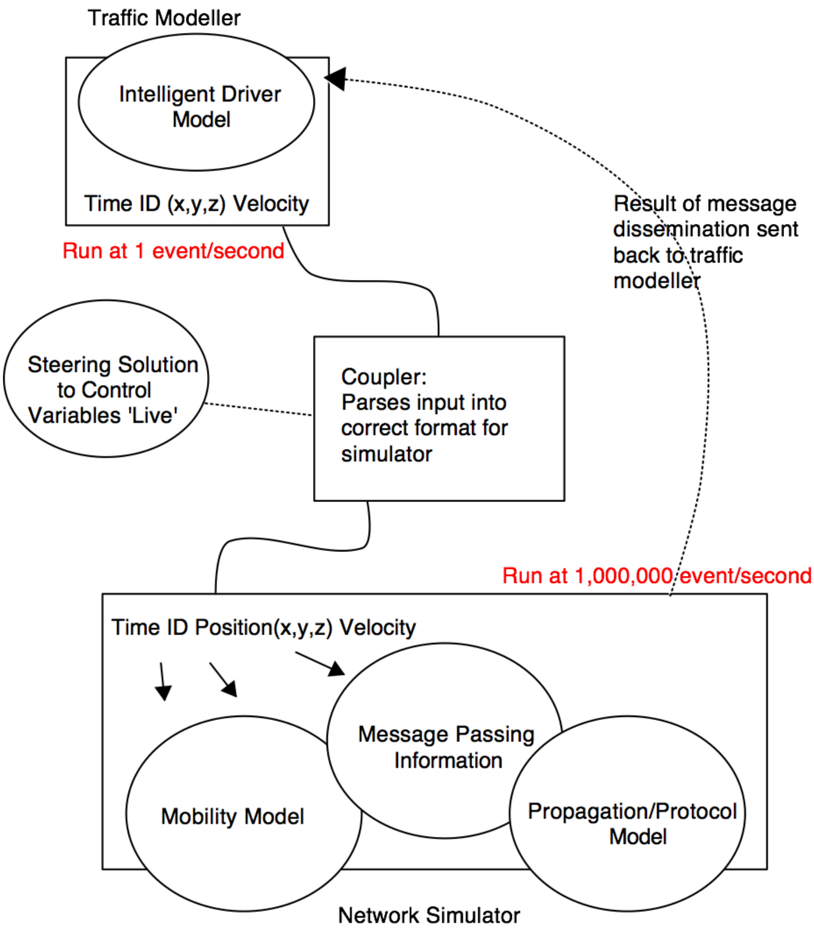


FIGURE 4.2: Synchronisation and coupling functions as a pseudo-flow chart

would not be suitable for all ITS applications, but would be ideal for collision avoidance systems, where only nearest neighbour information is pertinent.

4.4 Parameter Selection

The selection of parameters to vary in a simulation depends on the results required, and how these can be used to alter the settings of the network. The variance of connected parameters can yield similar results, for example reducing the transmission power and increasing the density could be hypothesised to cancel each other out, but in experimentation it is important not to over-fit the parameters due to expectation. Using the parallel methodology this becomes easy to avoid, by achieving simulations in far less time than a serial approach would allow.

In this chapter the vehicular density, packet transmission rate and packet size are varied, but there is also access available to change any of the parameters in the protocol stack and in mobility settings. In a vehicular environment the density and number of packets within reception range will be very dynamic, and the network efficiency will change. In order to build and establish a safety system a reasonable quality of service (QoS) is required, and this work formalises what could be expected.

4.5 Parallel Algorithm

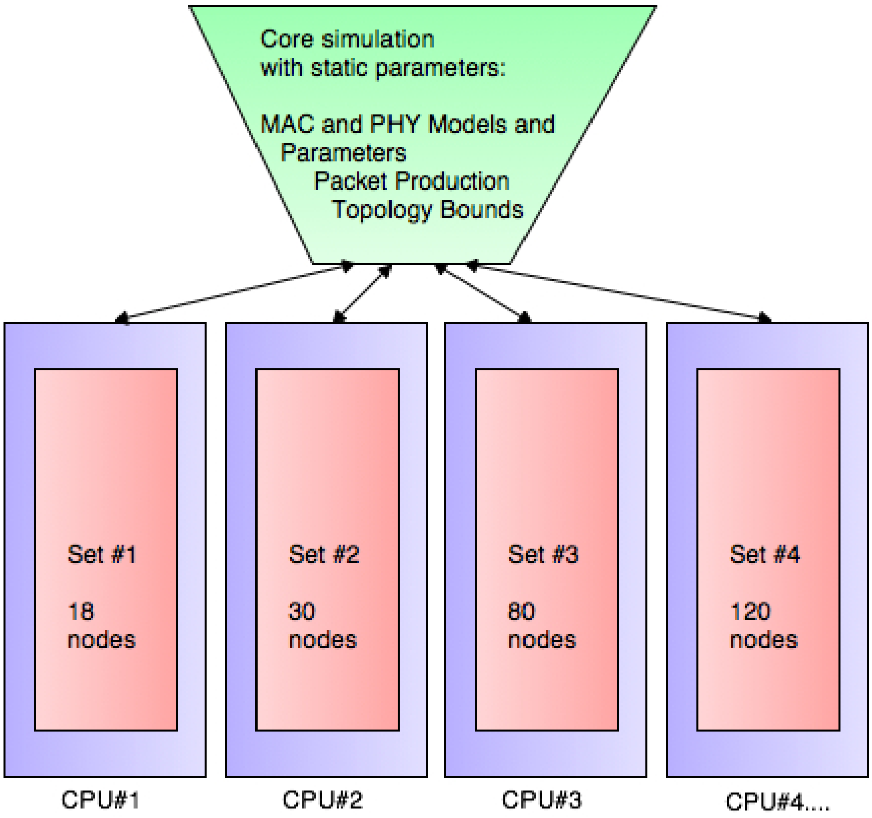


FIGURE 4.3: The master/slave spawning process

The algorithm that was designed to run the simulations with varying parameters uses a master and slaves approach. The master simulation creates the common settings and topologies for the slave instances to run. All of the parameters that can be varied are initialised and a function, that has information about the range of each parameter and an initial value (taken from the relevant specification) is used to select parameter 'sets'. These 'sets' are then created and spawned on a new processor. The results are communicated back to the master process when the simulation terminates, and the file output is created with a specific stamp, which contains the information about parameters selected and values applied.

The master process spawns new instances of the simulation in a flat hierarchy, and so the only communication is from the processor running the simulation back to the master processor. Due to the changing computational expense of each parameter 'set', the spawned simulations finish at different times, this reduces the overhead of communicating back to the master processor.

The following process shows how the simulations are initiated and spawned:

1. Configure universal parameters for ALL simulations
2. Select number of available processors
3. Specify variables for exploration

   For each variable {

      Calculate range and value of variable

      Assign parameters to a set

   }
4. Initiate master process

   For each process {

      Apply a parameter set of values to the simulation

      Setup file I/O

      Begin simulation

   }
5. As each simulation ends collect statistics
6. Once all simulations are complete perform post-processing
7. End Simulation

4.6 Simulations

The simulations are run in parallel for increasing vehicle density and packet transmission rate. The EU architecture documentation [60] for vehicle safety specifies a 2Hz rate for packet transmission, in normal operating mode. For a safety and collision avoidance application the production of packets may need to be much higher, up to 20 packets per second. In the following experiment measurements are taken of the packet delay that occurs (i.e. the time taken from a transmission request to the reception at the MAC layer of the receiver(s)) when the parameters are varied.

The simulation covered 20 densities and 5 packet transmission rates, for 5 packet sizes. The average realtime for an hour of simulation time is 7 minutes, which equates to nearly 2.5 days on a serial processor. This method can complete the simulation runs across 500 processors in 15 minutes (subtracting the time to compress and copy back the results).

The results show that packet delay increases as a function of density and packet size, but that at these packet transmission rates, there is little change. This is due to the large number of events that can occur in a wireless network each second.

Example: Packet Size 200b on a 6Mbps channel.

$$T_{P_{200B}} = \frac{P_{size}}{Chan_{bandwidth}} = \frac{1600}{6,000,000} \tag{4.1}$$

$$T_{P_{200B}} = 2.6666666667 * 10^{-4}s = 0.266ms \tag{4.2}$$

This shows that the channel is capable of carrying 3750 packets per second. At the highest density shown here (209) that would require each vehicle to transmit 18 packets per second, and be in range of every other vehicle. This level of transmission rate is achievable by modern network devices and in the 802.11p bandwidth allocation.

4.7 Challenges and Performance Analysis

The simulation of VANET requires complex processing that occurs through 'events' in the simulation that operate according to a list, or schedule, that is centrally controlled. The processing time required for a simulation (described and compared between simulation software in Section 2.4 on page 33) is the time it takes for the scheduled list of events to complete and the results output to file.

The full parallelisation of network simulation software has been attempted by several systems, including QualNet (a commercial offering described on page 43) and OMNET++ (page 41). One of the main methods to make serial simulation software parallel is to enable packet federation, which farms out the processing required to send and receive packets onto individual processors. Packet federation allows the major processing in a

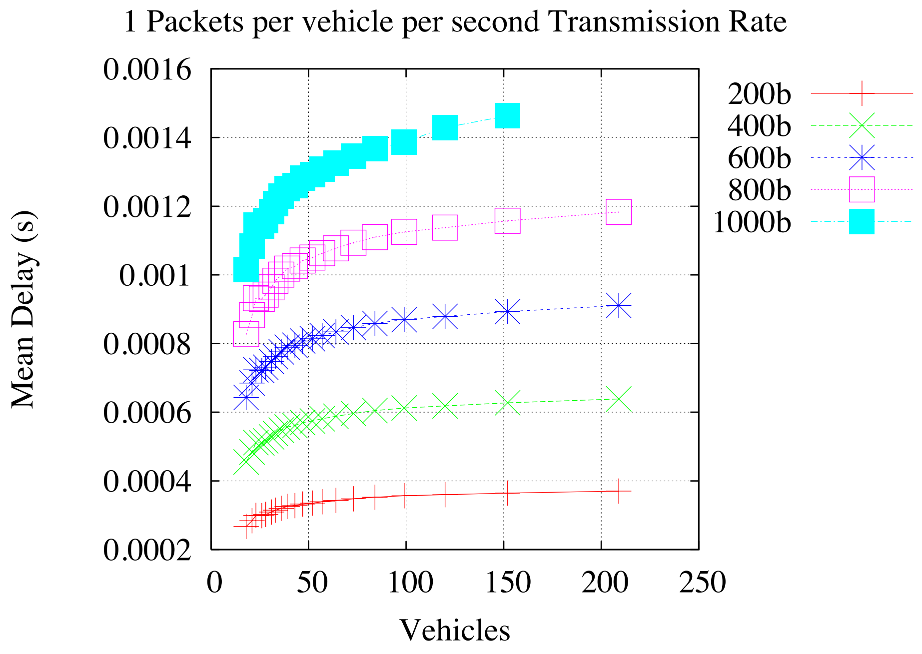


FIGURE 4.4: Delay with transmission rate of 1ppvps

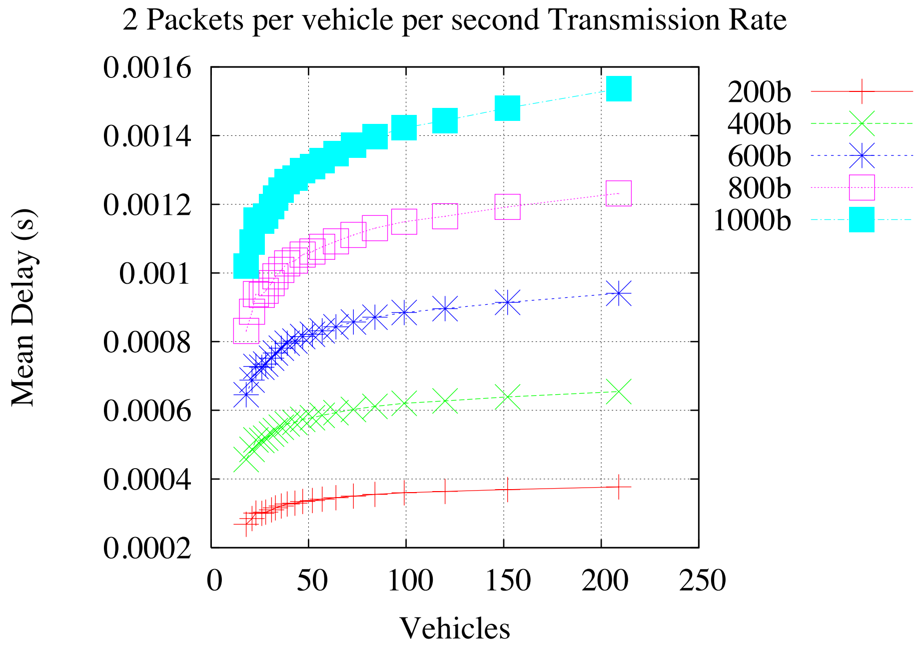


FIGURE 4.5: Delay with transmission rate of 2ppvps

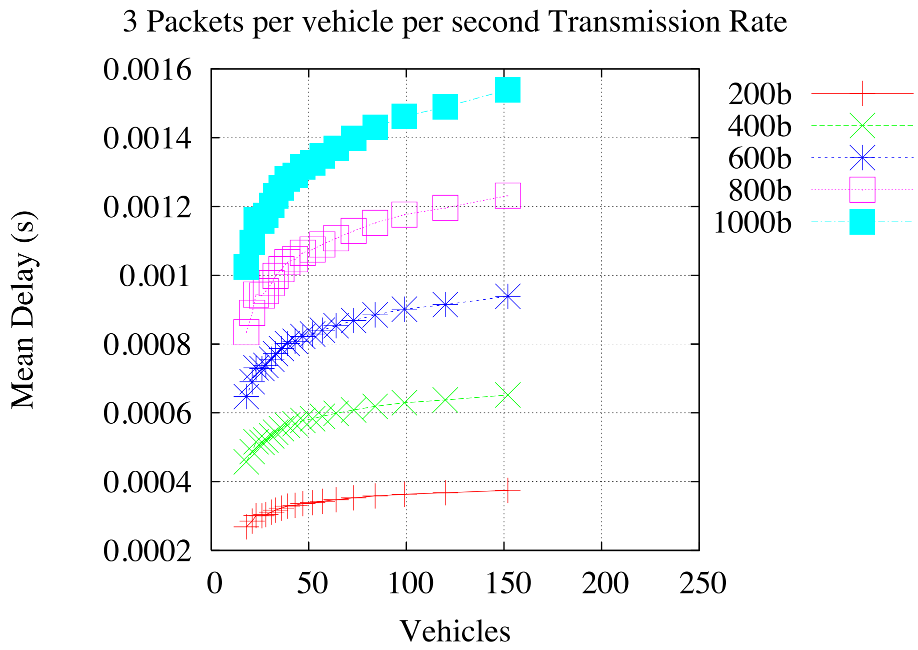


FIGURE 4.6: Delay with transmission rate of 3ppvps

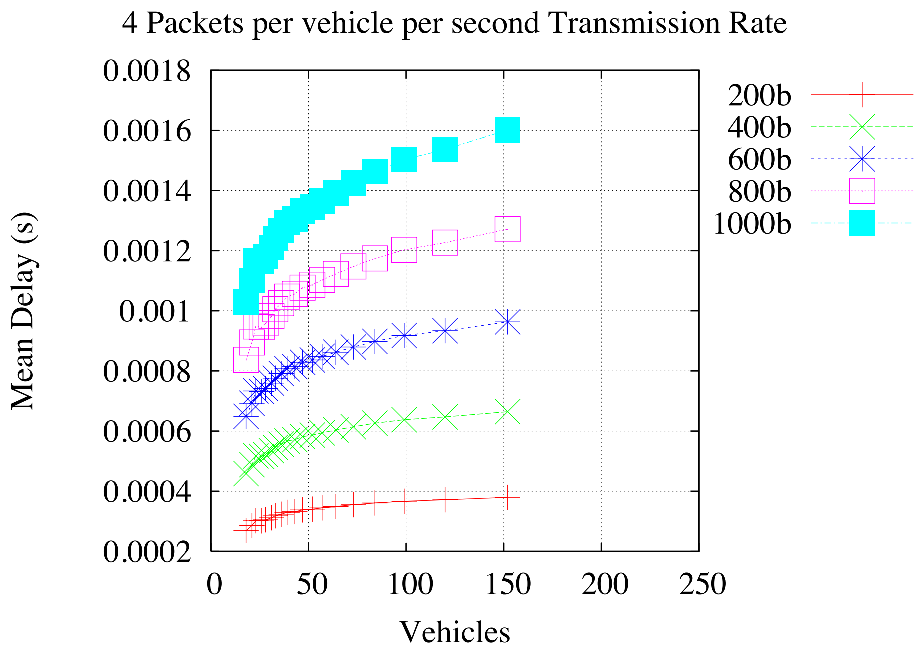


FIGURE 4.7: Delay with transmission rate of 4ppvps

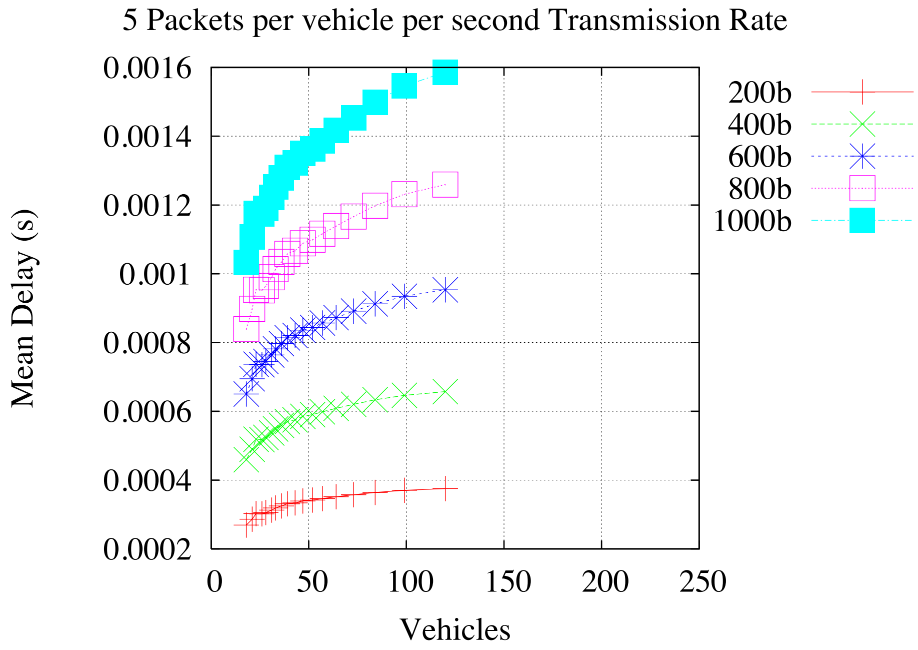


FIGURE 4.8: Delay with transmission rate of 5ppvps

simulation to be shared between interlinked processors. The complex nature of sending and receiving packets does create the necessity for a large amount of inter-process communication to achieve efficient packet federation, which can create a bottleneck or slowdown in the overall simulation.

In order to enable operation on a more diverse range of processing systems, from grid to cluster and multi-core, the method proposed in this chapter offers an better method to explore VANET systems in a shorter time than by running in serial. The algorithm developed offers a different approach to performing large-scale and high-fidelity simulations in a shorter time than through serial processing.

4.8 Conclusions

The work presented here shows how the parameters that vary in a vehicular network can be searched, using a parallel algorithm, in order to experiment with many scenarios and applications, in a shorter time than if it were run on serial processors. The results obtained can be used to model how a vehicular network will operate and what parameters achieve the best network efficiency. This information can be used to make decisions in realtime intra-vehicle and also to design and adapt the network equipment that manufacturers will install in their products.

The parallel algorithm defined in this chapter can scale to many thousands of processors, with only a small overhead for communication and file input/output.

Chapter 5

Universal Performance Behaviours in Vehicular *ad hoc* Networks

The challenges facing VANET are greater than those of a traditional wireless network or MANET, due to the more demanding environment in which vehicles operate. The high velocities of vehicles mean that link times, where data can be sent between two or more nodes, can be very short. Due to the safety-critical nature of the VANET ITS applications and the costs involved large-scale experimental evaluation of these networks is not a viable option. Therefore, it is of great importance to use high-fidelity computer simulations in order to both explore the remaining challenges of the technology and to develop improved protocols and algorithms which ensure successful performance of VANET under highly stringent requirements on reliability, speed and scalability that are imposed by most ITS applications as shown by Blum et al. [82]. Unfortunately, the current body of research on computer simulation of VANET has been hampered by the computational effort required in order to consider all of the important factors that influence the performance of highly dynamic networks created by thousands of vehicles which communicate wirelessly as they move in road networks. These factors include the characteristics of the underlying vehicular traffic (mean densities and velocities) which dictates the network topology and link lifetimes of the resulting vehicular networks, as well as wireless communication parameters such as transmit powers, data transmission rates, and packet sizes. Consequently, to our knowledge, current simulation studies of VANET have been limited to exploring the performance of these networks only within a relatively small subset of the above multi-dimensional parameter space.

There are several methods by means of which data can be transmitted and propagated through a VANET, sometimes called communication primitives. The main method studied in this work is single-hop broadcast to nearest neighbours, where data is sent to vehicles within range and acted upon there. Other primitives are applied depending upon the specific requirement of the ITS application. The single-hop primitives are usually broadcast or unicast, whereas multi-hop transmission can use multiple casting methods such as receive and forward all received data packets, receive-store-forward or receive and forward according to predefined rules (Nekovee [83]). Little and Agarwal [84] provide a multi-hop approach to information propagation in VANET that could also be applied to this scenario if delay tolerance capabilities are required that is also rule-based.

As discussed in Chapter 2 (on page 19), VANET applications are often referred to as dedicated short-range communications (DSRC) as specified by the European Union (EU) [85]. This complements the work already done in the US on DSRC so that vehicle manufacturers can design VANET to operate in either region. To this end, the EU has also reserved 70MHz (3, 10MHz wide channels for safety-of-life applications and 4, 10MHz channels for control and other uses) in the 5.9GHz spectrum for DSRC and Vehicle-to-Vehicle (V2V) applications [86], as shown in Fig. 2.3. VANET can also be referred to as wireless access in vehicular environments (WAVE), under the specifications of the IEEE 1609 draft standard [87].

In this work we use novel algorithms that exploit a large number of computer cores on a substantial computing cluster, each performing one high-fidelity simulation instance of a vehicular network formed in a highway with a different set of parameters. The full range of parameters we study leads to up to 2000 simulation instances, which would take approximately one year to run on a single processor machine. Using the very large set of simulation data obtained we are able to perform, for the first time, a comprehensive study of the performance of vehicular *ad hoc* networks in the four-dimensional parameter space which is formed by vehicular density, ρ, the size of data packets transmitted, S, the packet transmission rate, f, and the transmission power, P. Subsequently, we examine the functional dependence of two key performance metrics for VANET, single-hop packet reception latency, τ and single-hop packet delivery rate, η, within the above parameter space. Our analysis shows that both τ and η do not depend on individual parameters of vehicular traffic density, packet transmission rate and transmit power, and packet size but instead can be written as a function of packet size and a new composed parameter, the so-called communication density [12]. This finding dramatically simplifies computer simulation studies of the performance of large-scale VANET since, instead of having to explore a four-dimensional parameter space, a much smaller set of simulation scenarios for different communication densities and packet size needs to be performed. Finally, analysing the behaviour of τ as a function of communication density we find that this quantity shows a critical behaviour as a function of communication density with a well-defined phase transition at a critical value of communication density below which it manifests a power law dependence on communication density, with a *universal* exponent which is independent of the transmitted packet size. We show that the universal behaviours of VANET, which are revealed in the results, could be be used as the basis for design of new adaptive communication protocols in future applications of VANET in intelligent transport systems.

5.1 Concept

The modelling and study of vehicular *ad hoc* networks using computer simulation allows a great number of scenarios and situations to be studied. In this chapter the system for simulation, which can study the massive dimensional space of available parameters, is presented. From the intensive analysis of the results, both fine-grained and high level, universal behaviours in the network have been discovered. The universality of packet delay and success rate that has been found means there are parameter 'sets' (i.e. a collection of parameters in a simulation instance) that equate to the same average end-to-end latency and create the same level of packet success.

5.2 Simulation Methodology

In this section the models that make up the simulation system are described. The network simulator we used was NS-3, implemented by Lacage and et al. [81], which contains many models for MAC/PHY layer types and several propagation models. In a computer simulation it is important to use models that accurately reflect the system we are interested in; so by using established software and incorporating validated models we can ensure an appropriate level of realism and complexity.

5.2.1 Vehicular Mobility Modelling

The modelling of vehicular mobility is primarily concerned with the movement of vehicles on a simulated road and what velocity and heading changes they will make. This approach can be broken down into a plethora of components, from driver type to vehicle age and condition.

The environment in which VANET operates means that minor changes to operational parameters and settings can lead to large effects. Vehicles moving at high speeds will have a short opportunity to share data with each other, so the modelling of these movements is very important to our work. The link times in a VANET scenario are calculated as a function of velocity from the relation shown in Eq 2.1 (page 24).

The mobility modelling specific to our research draws from the work of Treiber et al. [44] whose model was developed from analysing the motion of vehicles in a limited roadway (i.e. a highway with few on/off junctions) as described in 2.3.1.

5.3 System Parameters and Performance Metrics

The study of network performance requires that a number of parameters are explored, and a number of metrics are used to measure the effects on the network and the performance of V2V communication primitives. In this section we present the parameters we vary, how they interact and what effect they have on the network utilisation and operation.

5.3.1 Parameters

The parameters we vary in order to study the network performance are distinct from the parameters of the MAC and PHY layers, because these are normally specified by the standards (i.e. 802.11p), and most devices (network interface cards, wireless adapters etc.) do not allow the tuning of these settings, other than switching between 802.11a/b/g/n etc.

5.3.1.1 Vehicular Density

The density of vehicles will affect the channel utilisation and medium contention in a VANET. Given an average vehicle length of $4m$ (this covers most cars and small vans, but not motorcycles or trucks), the maximum number of vehicles per lane in a $1km$ stretch of road would be approximately 250, but this would be static with vehicles bumper-to-bumper. The maximum number of vehicles in a given field V_{max} is:

$$V_{max} = \frac{L}{l} \tag{5.1}$$

where L is the length of the road (m) and l is the average length of a vehicle (m).

Generally, for traffic in motion, the density will have to be much less than the maximum given in Eq. 5.1. However, for a VANET to arise, at least two vehicles must be within range of one another, so there is also a minimum number of vehicles in the field V_{min} to initiate communication:

$$V_{min} = L/Tx_{range} \tag{5.2}$$

for a given transmission range Tx_{range} (m) and a road of length $L(m)$.

The density of vehicles is given by $\frac{V}{L}$. The maximum density (V/m) is given by:

$$\rho_{max} = \frac{\frac{L}{l}}{L} = \frac{1}{l} \tag{5.3}$$

and to ensure communication is possible, a minimum density (V/m) of:

$$\rho_{min} > \left(\frac{1}{Tx_{range} - l}\right) \tag{5.4}$$

The minimum density is strongly related to the transmission range, which is a function of the transmission power, discussed later in this section.

5.3.1.2 Packet Size

The size of data being sent across the medium will affect the channel utilisation and backoff function of the MAC layer. The payload (the actual data being sent, without the encapsulated headers) is defined by the application running the transmissions. As shown in the background section (Section 2 on page 19, the majority of applications send at least the following information:

- GPS Location: 6-8bytes (2B for the degrees and 4B for the minutes and seconds)

- Vehicle ID: 32bytes (for a unique and identifiable number)

- Bearing: 2bytes

- Velocity: 2bytes

So we can see that the bare minimum of data that can be meaningfully sent is approximately 44 bytes. Given the need for more information in modern devices, we can expect there to be more demanding data requirements as the technology evolves, greatly exceeding the minimum required.

The contention for the medium is based on the amount of time required to utilise the medium for data propagation. A larger packet will require longer time within the medium and increase the contention (up to the maximum allowed size before it is necessary to fragment the packet) for access. Therefore, the performance of the application must be balanced between the data requirements and the required access to the medium.

5.3.1.3 Transmission Rate

Directly related to the size of the packet is the rate at which it is transmitted. The Euro D31 project specifies that vehicles send out regular 'heartbeat' messages at a rate of 2Hz [60]. In order to avoid vehicular collisions in an emergency situation, however, a much larger number of transmissions could be required, not only to send out more data but also to tolerate the losses in a highly contended network [88]. Torrent-Moreno [89] produce a position-based message-forwarding strategy that could be applied if the system was multi-hop enabled, but in this simulation the single-hop packets are sent according to a Poisson distribution.

Most DSRC and WAVE technology applications suggest a maximum data rate of 6Mbps using OFDM, so a channel is capable of carrying and holding a large amount of data, in VANET and ITS terms. Part of our work intends to show how the network performs when put under load and how well data can be disseminated when the network load is light.

An example of bandwidth utilisation is as follows: A 6Mbps (6,000,000 bits per second) channel can hold 750kBps (6,000,000 / 8, bits to bytes). An average packet size is 200B with approximately 50B for headers, encapsulation and tagging. This means that we could send 3,000 packets within the medium per second, a very large number. But if we imagine a highly congested highway where the average number of neighbours was 110, the fully utilised channel would occur at close to 30 packets per vehicle, per second. Including backoff functions, interference and overlapping slot synchronisation it would be easy to reach a fully utilised network, and incur high packet losses and large end-to-end latency.

5.3.1.4 Transmission Power

The power at which a packet is transmitted, expressed in Watts or decibels per metre (dBm), will influence the distance over which it can be usefully received (useful referring to a reception power above a minimum to extract the signal). It would appear that increasing the transmission power to a large value would increase the probability of reception, but due to the contention for medium access in VANET, this may lead to a network blockage.

The maximum permitted transmission power is defined by the FCC in the USA and by ETSI in Europe. DSRC and other ITS applications are generally used in the 5.9GHz ISM spectrum. This region is licence free as long as the maximum EIRP (effective isotropically radiated power, a value of the desired transmission power plus antenna gain) is adhered to. The power levels we study are:

13dBm = 19.95mW - Minimum power most 802.11 devices can transmit

17dBm = 50.11mW - General operating power for home Wifi (802.11a/b/g/n)

20dBm = 100mW - Maximum EIRP allowed by ETSI

24dBm = 251.9mW - High power for long distance communications (e.g. WiMAX)

We should note that the maximum EIRP allowed by the FCC in the US is 36dBm, or 4.00W.

5.3.2 Performance Metrics

The way in which we measure the performance of a VANET is important. There are many aspects and functions operating in the simulation system, at very small time steps, across each individual node and over the network as a whole. In this section we present the metrics analysed during a simulation.

In a highway environment when we are simulating vehicular network operation the two main parameters we will explore are the end-to-end latency of a packet travelling across the medium, and the success rate of packets sent. Vehicles that transmit data onto a shared medium will have to contend for access; that data will take time to leave the transmitting station and be received at the receiver. The application of VANET to safety systems relies on data being sent and received within a timeframe, set by the specific application, and so analysing the time it takes individual packets to be sent and received explores the operating parameters of these applications. For example, a collision avoidance application (such as that shown in Fig. 5.1) requires the packets, containing location and heading data, to be sent and received with a very low latency (below 100ms could be acceptable) and high success rate (above 70% for example), so that vehicles have time to compute a safe velocity and heading to avoid the vehicles around them. The rate of packet success and latency of data in a collision avoidance application are proportional to the velocity and density of the vehicles involved.

The packet success rate is measured by vehicle and by simulated field and denotes the number of packets that are sent and those which are received successfully and unsuccessfully. This metric really provides the quality of service that can be attained in a given scenario. ITS and safety applications will have to rely on an appropriate level of packet success that we can ensure is reached through our simulations.

We use simulation to examine the functional dependence of each metric (R) as a function (F) of the parameters varied:

$$R = F(\rho, s, f, p) \tag{5.5}$$

where ρ is the vehicular density (vehicles/m), s is the packet size sent in bytes, f is the transmission rate (Hz) and p is the transmission power (dBm).

Generally speaking, each performance metric (in R) will be a multi-variate function of the parameters investigated. We use the results of our simulations to map out the functional dependence of each metric considered and to make the relationship explicit.

5.3.2.1 End-to-End Latency

The time it takes a packet to leave the destination application, pass down the protocol stack, cross the medium and then be received, is important in calculating how many packets must be sent to disseminate data in a timely manner. When the network is lightly loaded the end-to-end latency is expected to be the propagation time plus a small amount for processing in the device circuitry. However, once the network becomes well-utilised, a number of delays can occur to the data packet. These delays could lead to loss of data and lack of timeliness in data reception. In applications such as collision avoidance, V2V communication is used in order to alert vehicles of a sudden breaking by vehicles ahead instead of relying on sight and acknowledgement of the brake lights. The end-to-end latency of alarm dissemination is therefore a crucial parameter in assessing whether the V2V technology is capable of meeting the stringent performance requirements of safety-of-life applications. The simulation of V2V networks allows us to push the limits of vehicular communication and assess the performance of the network in different scenarios.

In order to measure the end-to-end latency (T_{e2e}) of a packet in NS-3, we use the user-contributed delay/jitter estimation code. This operates by applying a tag to the packet as it passes through the protocol stack with a timestamp, that is then read upon reception. This measures the time between the packet arriving at the MAC (top) layer of the sender and the arrival at the lower MAC layer of the receiver. At this point the measurement includes delay in enqueuing to the MAC/PHY transition and backoff delay, including the time for propagation across the medium.

$$T_{e2e} = T_{TxMAC} + T_{TxPHY} + T_{medium} + T_{RxPHY} + T_{RxMAC} + L \tag{5.6}$$

where T_{e2e} is the end-to-end time, T_{TxMAC} and T_{TxPHY} are the time in the sender's stack (including DCF backoff and queuing time), T_{medium} is the time taken to propagate across the medium, T_{RxPhy} and T_{RxMAC} are the time spent in the receivers stack and L is the electronic loss in the circuitry.

For each simulation scenario and each parameter set we record the end-to-end latency of every successfully received packet. These values are collected during simulation runtime along with the total number of successfully received packets which gives us our mean average. The distribution of these latencies is often head-heavy, in that most values occur in the lowest possible time, as described in the previous paragraph but with no delay awaiting medium access or queuing in the OSI layers of sender or transmitter.

From an ITS application perspective, the end-to-end latency of a packet is an important factor so taking accurate measurements (in NS-3 this is at the nanosecond scale) is key to analysing the application performance in varying network loads. Exploring different packet sizes, vehicular density and transmission rate can exploit this metric for network utilisation and application efficiency.

5.3.2.2 Packet Success Rate

The success of packet transmission is one of the most important metrics for studying the performance of a vehicular network, as the success rate determines the operational limit of a given ITS application.

The success of a packet being received depends on many variables, propagation of radio waves, error in reception and collisions within the medium. The following is a brief explanation of each:

1. *Propagation Loss* - In NS-3 there are two settings in the calculation of packet reception. The energy detection threshold (EDT) defines a threshold of received power below which the signal cannot be recognised above background noise, and the CCA1 threshold (see Learmonth and Holliday [29] for more details of CCA), below which the physical

device cannot switch from idle to CCA mode 1, to begin capturing data. This means a packet must be received at a sufficient power to capture any data.

2. *Reception Error* - As data is taken off the medium, there is the possibility that some bits are collected in error. If this bit error rate (BER) becomes too high, the entire packet will be dropped. Occasionally, the packet may be corrupted or damaged in transit through the protocol stack.

3. *Packet Collisions* - Packet losses may occur due to the so-called 'hidden node' effect. This causes corruption taking data off the medium when the physical layer synchronises to a reception event which is corrupted by a second event (another packet on the medium). In *ad hoc* networks it is difficult to avoid this kind of collisions.

5.3.2.3 Communication Density

The concept of communication density was put forward by Jiang et al. [12] as a means to compare different utilisations of vehicular networks. They define the communication density as the product of transmission rate, transmission range and vehicular density

$$\Upsilon = \rho_{km} \times l \times p \tag{5.7}$$

where Υ is the communication density, l is the transmission rate (Hz), p is the transmission range (m) and ρ_{km} is the vehicular density (veh/km). Communication density furnishes us with a very useful metric to describe various scenarios, which can then be used to compare network properties described by other parameters.

The communication density adequately displays the links between the three factors comprising the function. For example, an increase in vehicular density and a reduction in transmission rate could lead to similar network utilisation. In our results (Section 5.5) we show plots for data with and without utilisation of the communication density parameter.

Parameter	*Value*
No. Vehicles	90,105,115,130,140,155,165,180,195,215, 235,260,285,320,365,420,495,600,760,1045
Propagation Model	Two-Ray Ground Reflection Model
Propagation Lambda	0.058 ($\frac{C}{f}$)
Antenna Height	1.5m above Vehicle Z coordinate
Packet Rate	1,2,3,4,5,10,20 per vehicle per second
Packet Size	200b,400b,600b (payload)
Tx Power	13,17,20,24 dBm EIRP
RoadLength	10000m
Simulation Time	600s
Simulation Runs	10 per parameter set for averaging
No. of Parameter Sets	1,680
Total Simulations	16,800

Table 5.1: The model parameters for the simulated scenario, with expected values. As shown, the number of parameters explored creates a large number of simulation instances which are readily explored using high performance computers and parallel computing algorithms. Transmission power (Tx) is measured in equivalent isotropically radiated power (EIRP). The propagation lambda is the result of frequency (f) divided by the speed of light (C) and is used to calculate the wavelength of the transmitted signal.

5.4 Simulation Scenario

The scenario used for obtaining network performance statistics is described in this section, with clarification of the reasons for the particular details of the setup. The scenario is based on a section of highway, with two lanes of traffic at varying densities using the IDM developed by Treiber to compute the vehicle positions. Fig. 5.1 shows the simulated scenario in an illustrative way (i.e. not-to-scale), and the possible application of collision avoidance. The distance that LOS could operate after an incident is shown by the arrows in Fig. 5.1 and the ‘halos’ represent the distance that could be covered when V2V dissemination is used. The main model parameters varied are shown in Table 5.1, with the range of values we explore. The number of vehicles represents an increasing density of vehicles on the road section simulated, leading to an increase in the number of possible neighbours and therefore an increase in network utilisation. The average number of neighbours for each transmission power is shown in Fig. 5.2. As can be seen from Fig. 5.2 the average number of neighbours in a simulation increases linearly with the number of vehicles, and therefore vehicular density, with a coefficient which depends on the transmission power.

In our simulations, the two-ray ground reflection model achieves an adequate representation of the real propagation of radio waves. At the transmission powers studied, only $13dBm$ falls below the crossover distance, where the Friis free-space calculation (explained in the work of Hogg [27]) is used for modelling the pathloss exponent. The two-ray ground model does not include calculations for shadow fading, which is the calculation of the influence of all other interferers (i.e. transmitters) on the transmitted signal. Shadow fading is accounted for in NS-3 with the measurement of the noise on the channel as part of the DCF functionality, which is included in the calculation of pathloss between two points, the sender and the receiver. This helps to calculate the bit error rate and packet error rate for received data, as per [74].

The packet rates studied increase from 1 packet per vehicle per second (ppvps) to 20ppvps, as this represents from half the specified rate in the D31 standard to 10 times the specified rate. The rate of transmission has a direct influence on the MAC layer calculations, the physical layer calculations, signal to noise ratio on the medium and packet success rates, so it is important to study a broad spectrum. The simulation of an ITS application therefore requires each station to send packets out at a given rate.

The payload requirement for most ITS applications is quite low, containing a timestamp, GPS location and other data (velocity, heading, etc.) which can be sent in under 500b. Obviously, as ITS applications become more complex, the data requirement will increase.

The length of the highway studied is 10km for all our simulations. A problem found in many simulations is the effect of edges, or boundaries, of the simulated area. If we imagine a straight road with vehicles distributed according to the IDM, then nodes at either end will only have a full complement of neighbours on one side, which is unlikely in a real road network. Several methods are available for dealing with this aspect of simulation. One simple method is to position the vehicles on a ring road, where the diameter of the circle is greater than the EDT threshold's useful distance. This would mean that the road length for our transmission powers would be much greater, requiring an exponential increase in vehicles, in order to keep the density the same. An alternative method is to wrap the ends of the road section around to join each other ($0 = 10000 = 0$), an approach which does not require many changes to the topology. The use of such periodic boundary conditions is exploited in a plethora of applications.

To deal with the finite size effects of edges on our simulations, which we found in a smaller system of 2km and up to 209 vehicles, we extended the road studied to 10km and re-plotted the vehicles accordingly. This increased the size of the system being studied by a factor of 5 but, due to well planned coding, only increased the computing time by a factor of two. The reason we did not choose to wrap the ends of the road around, or create a ring-road, as done by Killat and Hartenstein [90], is to reduce the possibility of vehicles becoming incorrectly located in the topology, according to the IDM. Once the vehicles are mobile the wrapping of the ends can lead to vehicles ‘jumping’ which will affect the results. The use of periodic boundary conditions in network simulation is complex and sensitive to errors, mainly because of the calculation of propagation pathloss between vehicles.

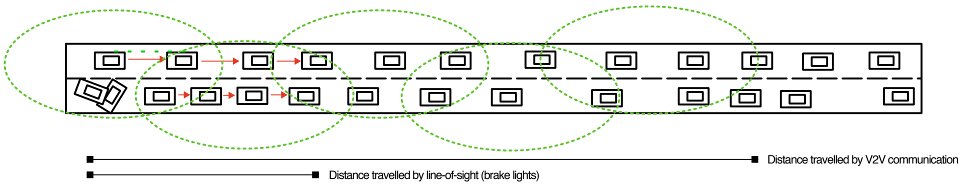


FIGURE 5.1: The scenario we are investigating. This highway situation and the resulting communication could lead to collision avoidance applications where wireless communication (shown as dotted halos around the vehicles) notifies drivers of an approaching incident much faster than line of sight and the visual cue of brake lights (shown by the arrows between vehicles)

The full range of parameters we study leads to 1,680 simulation instances, which would take approximately one year to run on a single-core desktop machine. In order to perform this many simulations in a reasonable time, a parallel processing algorithm was developed that completed the full parameter exploration in approximately 4 days. Given that each simulation takes between 1-4 hours to run, on a high performance computer with over 1,680 cores available, we can collect a full dataset in 4-5 hours. As the simulations become larger and more complex the amount of data that is collected must be post-processed and analysed, which will take more time. We run the bulk of the post-processing in simulation time in order to mitigate the requirement for data output to file, which is significantly slower than working memory (i.e. RAM).

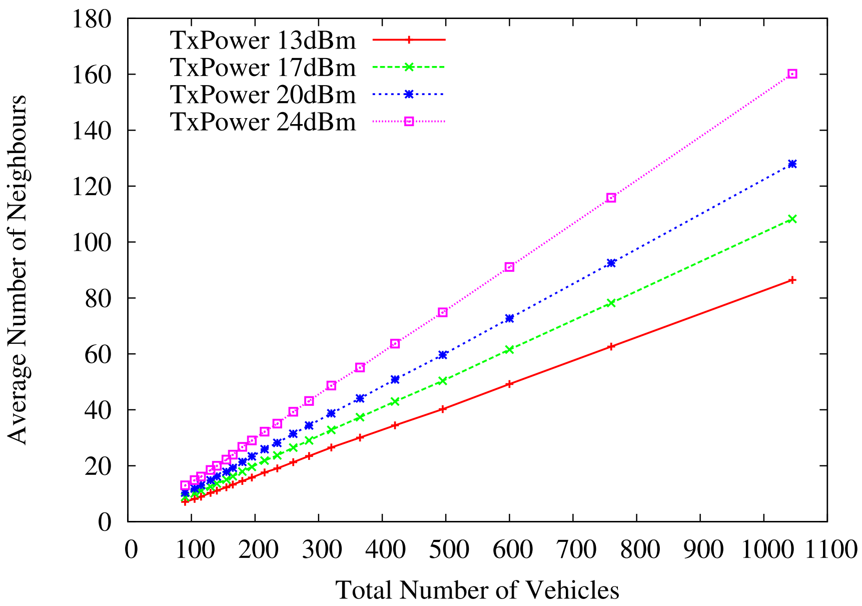


FIGURE 5.2: This figure shows the level of connectivity of a VANET in the simulations, defined by the number of possible neighbours a vehicle will have (i.e number of other vehicles within a usable transmission range),which is, in turn, a function of the transmission power.

5.5 Simulation Results and Analysis

In this section we present the results of the simulations that were performed based on the parameters introduced in Section 5.3 and showing the operational levels of our simulated system. The results have been produced from 4 full parameter explorations, using over 6,700 simulations. The metrics and performance indicators have been introduced in the previous sections of this chapter.

5.5.1 Network Connectivity

The study of the connectivity of a network with variable transmission power leads to the measurement of the number of neighbours per vehicle. Taken as an average for a particular parameter set, we can plot these values.

In Fig. 5.2 we show the linear increase in neighbours with transmission power and for increasing number of vehicles on a fixed-length road. We found that the functional

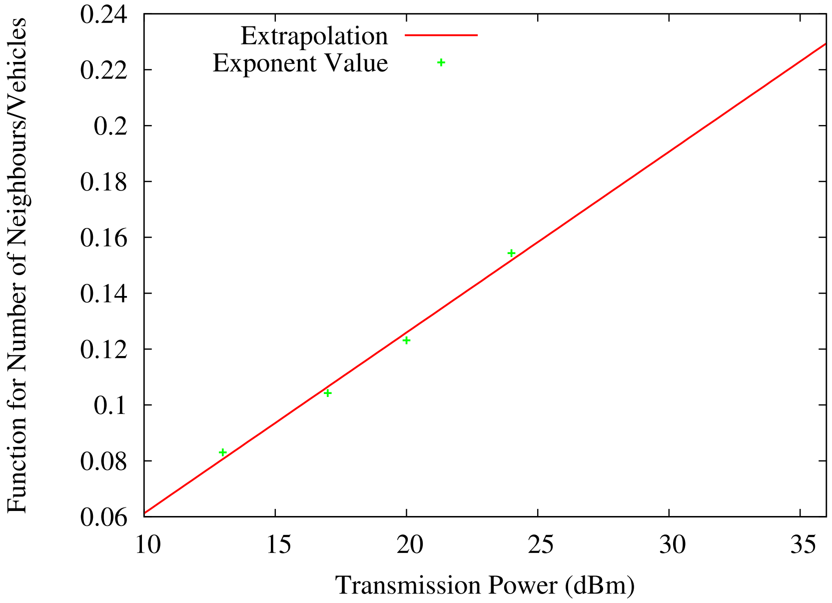


FIGURE 5.3: This figure shows that when we plot the exponent of each function in Fig. 5.2 we can derive a relation between connectivity and transmission power, and extrapolate this to a wider-range of transmission powers.

dependence of the average number of neighbours on the number of vehicles (N) and on the transmission power (p) is well described by:

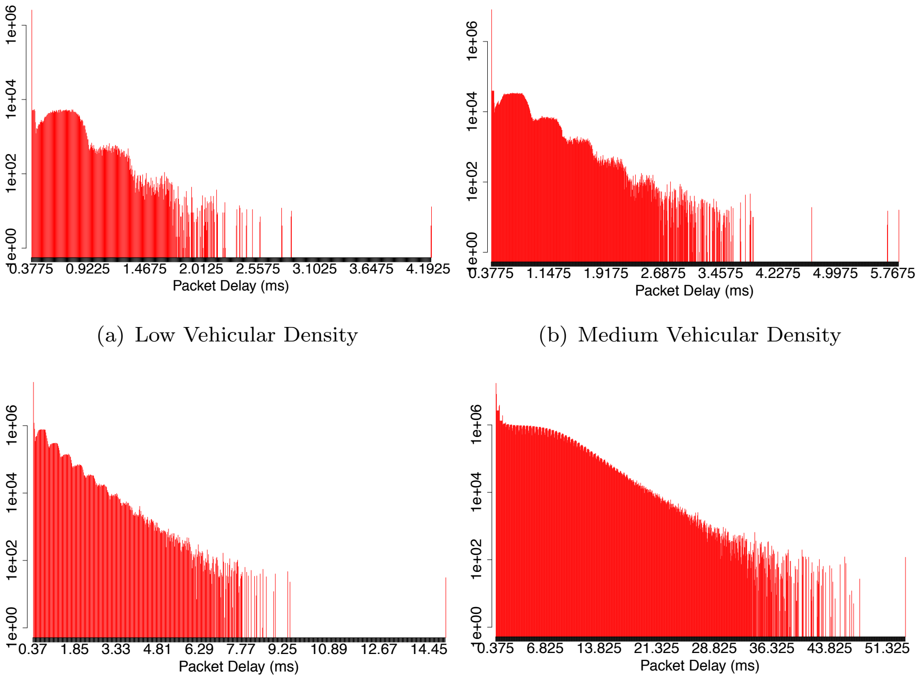
$$\bar{k} = \alpha \times p \times N \tag{5.8}$$

where $\bar{k}$ is the average number of neighbours and $\alpha = 6.47 \times 10^{-3}$.

Fig. 5.3 shows a plot of this functional dependence for increasing levels of transmission power.

5.5.2 End-to-End Packet Latency

As described in Section 5.3, the end-to-end packet latency is important to measure the timeliness of information dissemination in ITS applications. Boban et al. [91] formulated a method for calculating bounds on packet latency in VANET, and claimed that in highway environments a latency of up to 40ms is acceptable for most ITS applications. In our work we have explored a greater range of network utilisation than Boban *et al.*,



(a) Low Vehicular Density

(b) Medium Vehicular Density

(c) High Vehicular Density

(d) Very High Vehicular Density

FIGURE 5.4: These histograms show a scenario where the transmission power is fixed at 17dBm, the packet size at 200b and the transmission rate is 20Hz. The increasing density shows how the distribution of latencies in a simulation occur. The y-axis (frequency) is set on a logarithmic scale due to the large number of latencies recorded in the lowest bin.

from very light network traffic and vehicular density to a medium which is fully saturated with data. We present the results of these studies in this section.

Fig. 5.4 presents the distribution of latencies across a particular parameter set (as defined in the captions for the figures). We can see from these graphs how the distribution changes as the network utilisation is increased, but the majority of latencies recorded always remain at the lower (left) end of the histograms. From a) and b) in Fig. 5.4 we can see that the distributions show a repeating pattern as the latency increases.

5.5.2.1 Average Latency

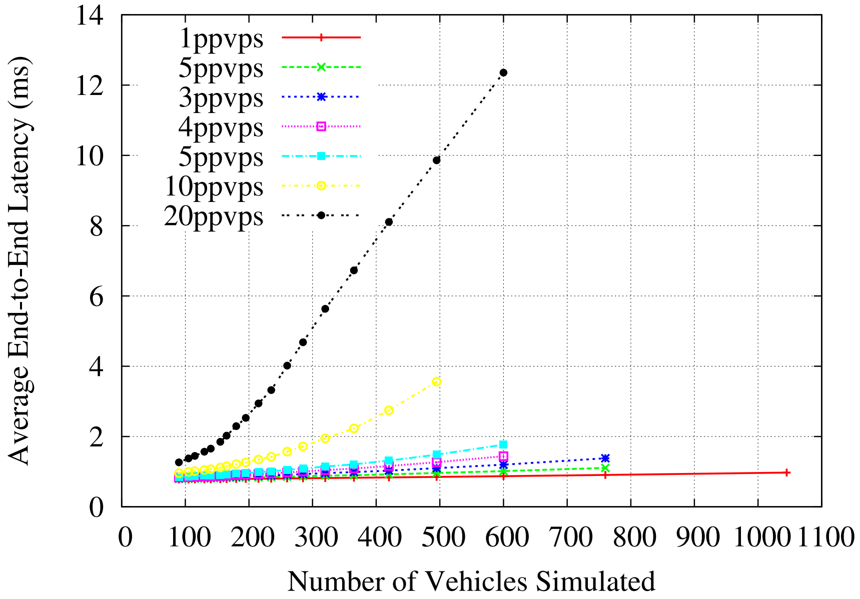
The average latency for a given parameter set shows how the latencies are influenced by transmission rate and packet size. When plotted as a function of vehicular density we can see that the variation between latencies depends on all the parameters varied. In Fig. 5.5 the comparison between using these independent parameters functions and the communication density is shown.

Fig. 5.5 shows the average latency for all the scenarios studied, as a function of transmission power, packet size and packet transmission rate. The end-to-end latency of each packet can be plotted as a histogram for each parameter set as shown in Fig. 5.4 which shows the distribution of packets across a specific parameter set. Exploring the distribution of latencies across a simulation can reveal more detail of the actual operation of the network.

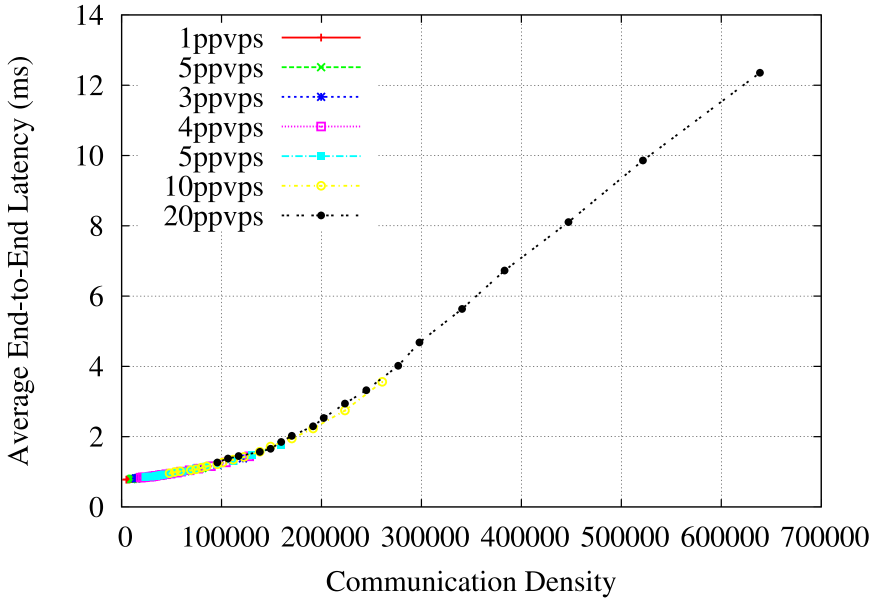
In Fig. 5.5 we plot the packet latency as a function of communication density which has important implications both in reducing the number of parameters that need to be explored and for the design of traffic-adaptive protocols, discussed further in Section 5.6. In Fig. 5.5 it can be seen that when plotted as a function of this quantity the curves all collapse into a single curve indicating that communication density is a universal function of τ and does not depend on independent variables.

The maximum latencies that can enable a particular ITS application are specific to the application. In the work of Boban et al. [91] a maximum of 90ms for an infrastructure-less VANET in DSRC conditions was derived. The authors investigated a small set of parameters for a specific application that is comparable to other work. We have investigated a much larger parameter space to explore the range where performance requirements are met and where they are not.

Fig. 5.6 shows, on a log-log scale, how the end-to-end latency of the packet in a network increases with greater use of the available bandwidth, and also greater contention for the medium. Each point in Fig. 5.6 is the average for a particular parameter set, equating to 588 points per packet size. When the log of the average end to end latency is plotted against the log of communication density, we see that the gradients for the different packet sizes are the same, until there is a transition to another exponential increase in latency. This behaviour is indicative of the maximum channel utilisation being reached by the larger packet sizes. The legend of Fig. 5.6 shows the derived exponent coefficients for the linear functions of each packet size, but omits the individual data files. For latencies below 11.2ms for 200b ($\Upsilon = 975000$), 36.5ms for 400b ($\Upsilon = 530000$) and 32ms for 600b ($\Upsilon = 350000$) the latency increases linearly, with shallow gradient. Above these values, however, we see a large increase in the average latency throughout a heavily congested and highly utilised network. We identify this behaviour as a transition from a free to a congested phase in V2V network traffic at a critical value.



(a) 200b Vehicular Density



(b) 200b Communication Density

FIGURE 5.5: The plots for average latency in a particular scenario are plotted in (a) against the total number of vehicles (over a 10km stretch of highway) and in (b) against the communication density for that scenario. Graph (a), of vehicular density, shows how the latency increases with transmission rate. In (b), which shows latency as a function of communication density, the latencies are seen to collapse onto a single curve.

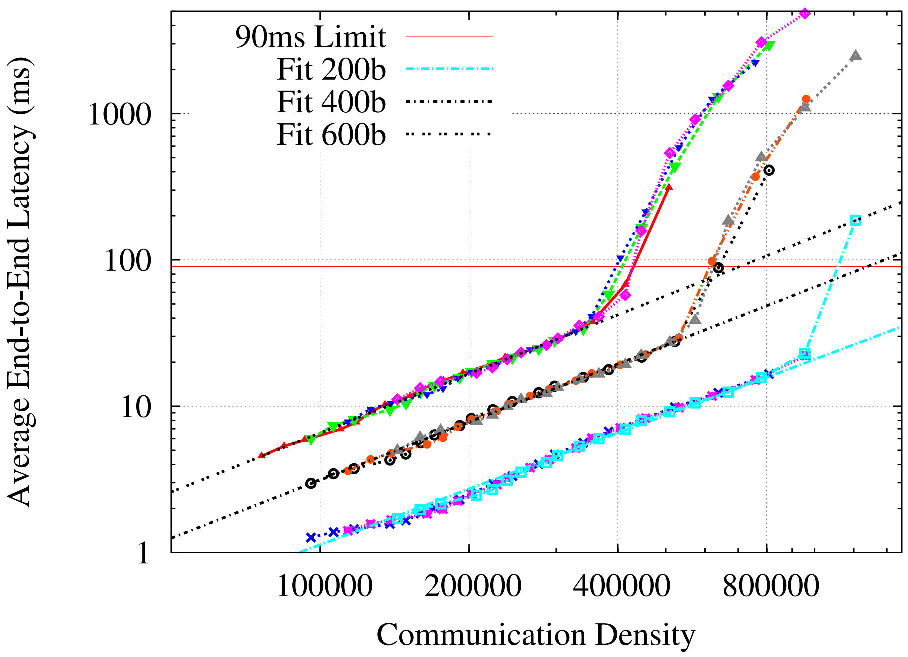


FIGURE 5.6: When the log of the average end to end latency is plotted against the log of communication density, we see that the gradients for the different packet sizes are the same, until there is a transition to another increase in latency. This behaviour is indicative of the maximum channel utilisation being reached by the larger packet sizes. The legend shows the derived exponent coefficients for the linear functions of each packet size, but, for presentation purposes, omits the individual data files.

The use of communication density (Υ) shows that the transition does not depend on vehicular density, transmission power and transmission rate separately, but is driven by a particular combination of these parameters.

The exponent shown in Fig. 5.6 is independent of packet size and is given by $\alpha = 1.3$ (with an asymptotic error of 0.6%). However, the critical value of the point where the behaviour changes to a higher gradient is a function of packet size. As described in Section 5.3, the communication density is the number of packets produced per second per kilometre of highway. As the communication density increases, so does the utilisation of the available bandwidth, to the point where the network is so highly contended that many packets remain in the outbound queues of the transmitting nodes (at both the MAC and PHY layers), leading to extremely high latencies. For each vehicle in a VANET there is a maximum capacity that the surrounding medium can hold which, when reached, leaves no space for new data to be transmitted. The 802.11 contention mechanism, combined with the asynchronous attempts for the medium in a VANET, mean that once a maximum channel capacity is reached, there may be nodes that experience difficulty in accessing the medium, because of the increased contention for the saturated medium.

Fig. 5.6 suggests that the end-to-end latency (T_{e2e}) is proportional to the communication density up to the point where the latency increases dramatically. From Fig. 5.6 we can state that the function of our performance metrics is reducible to a function of communication density and packet size:

$$\tau(\rho, s, l, p) = \tau(\Upsilon, s), \tag{5.9}$$

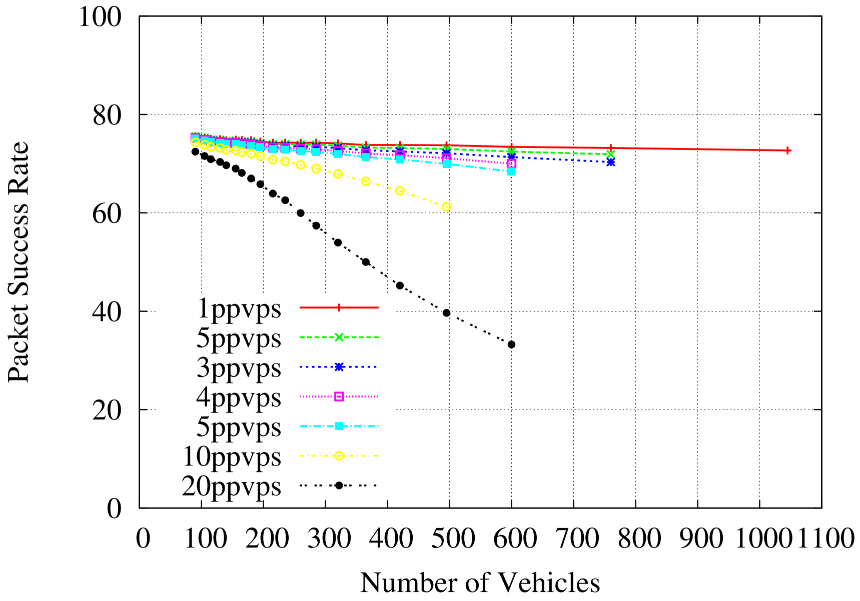
where ρ is vehicular density (stated in our work as the number of neighbours), s is the size of the packet, l is the transmission range, p is the packet transmission rate (Hz) and Υ is the communication density.

As Fig. 5.6 shows, the value of restricting data requirements in ITS applications, can produce a reduction in packet end-to-end latency. From the extensive studies we have performed, we see that packet size and communication density are important parameters to consider when designing an ITS or VANET application. In terms of sending data through a network, we can see how, at 200b packet size, over 800,000 packets can be sent in a kilometre of road with a reasonable latency. The latency of a packet is important for the timely transmission of data; the next section deals with the rate of successful reception, which can be more important in some applications.

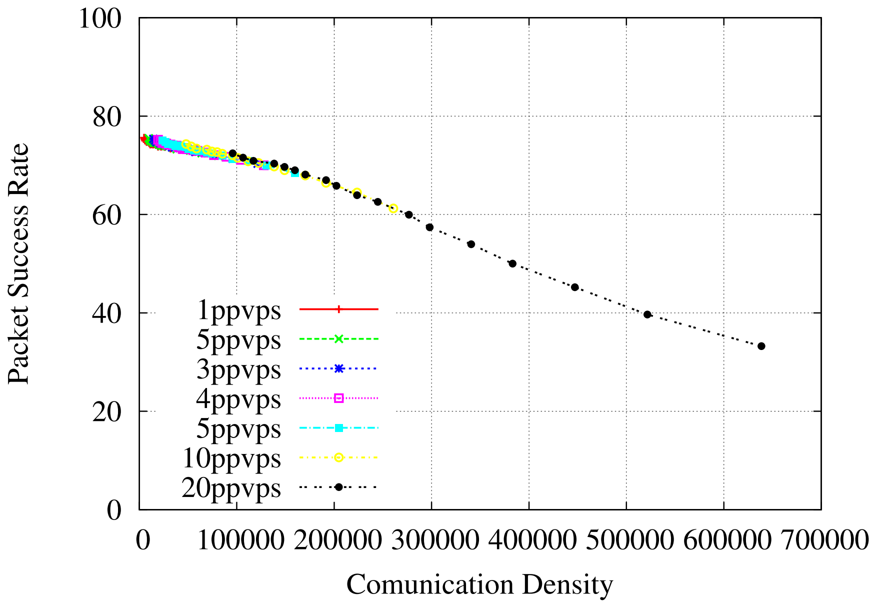
5.5.3 Packet Reception Success Rates

The success of packet reception is important for all ITS applications, to ensure that data is received when expected. As a network becomes more heavily utilised the chance of collision on the medium and loss due to signal error increases. In Fig. 5.7 we show the success rates for increasing utilisation as both a function of vehicular density (in number of vehicles on the road) and communication density.

Fig. 5.7 shows that for all the available parameters we have explored there are a number of different success rates, but that they group around packet size. The drop of all the packet reception success rates as the vehicular density increases is similar across the



(a) 200b Vehicular Density



(b) 200b Communication Density

FIGURE 5.7: This figure shows the packet reception success rates for our simulated scenarios. Plotting against vehicular density (a) we see that all the parameters studied lead to a variation in packet success, but when we plot the same data using communication density (b) the plots collapse onto a single universal curve

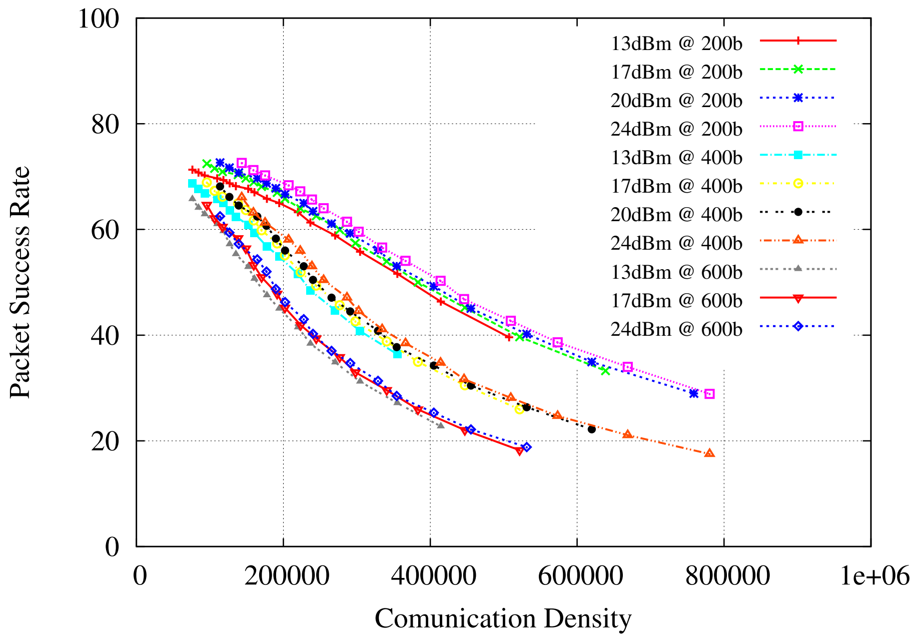


FIGURE 5.8: In this figure we plot the packet reception success rate as a function of communication density. We observe that the data for the three packets sizes collapse into three distinct curves, similar to the behaviour shown in Fig. 5.6.

parameter space, but it is difficult to see any universal behaviours for the simulations we performed. As can be seen in Fig. 5.8 the success of packets sent as a function of varying packet size and transmission power, when plotted against communication density, shows that the packet size is the only factor that distinguishes performance.

In Fig. 5.8 we plot the packet reception success rate as a function of communication density. We observe that the data for the three packets sizes collapse into three distinct curves, similar to the behaviour shown in Fig. 5.6. Despite the expectation that higher transmission power should provide for better packet reception success, we observe that this makes only a small difference when plotted as a function of communication density.

The main aspect of Fig. 5.8 is the lack of large change for different packet sizes onto a new gradient. At the communication density where the behaviour changes in packet latency (Fig. 5.6), there is no obvious change in the curves for each packet size. This confirms what was stated above, namely that the channel utilisation and saturation of the medium is the cause of this change. The reason we do not see such critical behaviour in Fig. 5.8 is that we are examining packets received, rather than the time taken to send a packet over a congested medium. In Fig. 5.8 we see how the packet success rate for three packet sizes drops as we increase the communication density. Each packet size that we studied shows a drop of 50% with increasing communication density, indicating that a highly utilised network with larger packets requires a greater resilience to packet loss.

5.6 Conclusions

In this work a comprehensive investigation into the performance aspects of vehicular *ad hoc* networks has been presented. The modern vehicle can be enhanced through the implementation of networking capabilities, but the proposed applications have to be stringently tested and examined to ensure adequate and reasonable operation. In our work we have explored a parameter-space that covers the range of the parameters and functions for the IEEE 802.11p draft-standard operations. This work has produced results that show network behaviours from which universal conclusions, applicable across the full range of parameters studied, can be discovered.

The 802.11p specification is still under development and the applications that drive the technology are being constructed rapidly. As is shown in Fig. 5.2, the connectivity of a network can be found by looking at expected neighbours for the different transmission powers. This result is fundamental to intelligent transportation system applications in respect of local knowledge, where decisions on transmission rate and packet size are variable.

We show that, by using the communication density metric put forward by Jiang et al. [12], we can collapse the average latency in a network onto distinct behaviours according to packet size. Due to computational constraints, Jiang et al. were not able to fully explore the communication density metric. We used high performance computing to achieve a complex network model, using the NS-3 simulator, and in a realistic timescale (1 day of computational work).

The difference between the behaviour of packet end-to-end latency and the packet success rate for a give scenario and parameter set show that even when the network parameters are widely different we may still find similarities in network behaviours. In the packet end-to-end latency we observe a sharp transition when the network medium becomes saturated, while we do not see similar behaviour in the success of packet transmission figures. The bounds discovered in our studies and the previous literature support the possibility of maintaining a high quality of service by implementing adaptive packet size and by monitoring the activity within the medium in the local area.

A next step is to explore the mobility in a reactive environment, where reception of data influences how vehicles move. This scenario may be used to explore collision avoidance for enhanced safety and requires a fully coupled mobility model. We also intend to use the results of this work to define an adaptive algorithm, to explore developing parameter changes in order to sustain a reasonable quality of service (in terms of packet reception success and latency) for a given scenario and application. This kind of algorithm will direct the parameters that are varied to maintain the quality of service and in order to demonstrate real-world applicability.

Our findings demonstrate that there are universal behaviours of vehicular networks with important consequences for intelligent transport system applications that should ensure their successful operation in a real-world environment.

Chapter 6

Optimisation of Network Performance in VANET

In order to operate well, ITS applications must sustain a high quality-of-service (QoS) in terms of packet reception success rate and end-to-end latency. As shown in Chapter 5, with certain network parameters and in a particular environment the packet latency and success rate can be calculated according to the communication density function. This means that if constraints are set for reception success rate and packet latency the network device can be tuned to keep within those limits, where possible.

The parameters and variables explored in Chapter 5 cover both the medium-access control and physical layers (of the OSI model in Fig. 2.1) and so to tune these parameters the network device takes advantage of cross-layer communication between the layers. The communication across the layers can be simulated using NS-3 in the same way as a real device, taking advantage of the encapsulation data that is added and removed at each layer of the OSI model. Cross-layer optimisation methods are well used in wireless technologies, such as Hendrantoro and Hendrantoro [92], where the high frequency (30GHz) wireless broadband system is optimised through manipulation of channel characteristics, such as OFDM transmission settings, in unison with MAC layer functions.

In this chapter a method for performing VANET simulations in NS-3 is presented, which uses a simple algorithm to optimise the network according to packet latency and success rate by adapting the packet size. The use of high-performance computing resources enables a quick exploration of the parameter space to choose the most optimised values for a given scenario. The application of this work can be applied directly to the devices that will operate in radio-equipped vehicles to ensure and maintain a high quality-of-service (QoS).

Parameter	*Value*
No. Vehicles	90,140,215,285,365
Propagation Model	Two-Ray Ground Reflection Model
Packet Rate	1,5,10,20 per vehicle per second
Packet Size	Variable (payload 100b-500b)
Tx Power	17,20 dBm EIRP
RoadLength	10000m
Simulation Time	10s

Table 6.1: Model parameters for the simulation scenario, as subset of those shown in Tab. 5.1.

6.1 Concept

In order to operate well, an ITS application must ensure that it achieves and maintains a high reception success rate and low packet latency. The ability of modern network devices to be finely tuned at the MAC and PHY layers offers the possibility to update the device settings in real time. Given a particular scenario an intensive exploration of the parameter space can build a lookup table of parameters to tune a network device and enable high QoS.

The parallel algorithms used previously in this thesis enable a fast exploration of a large parameter space, which will populate the lookup table and, using optimisation algorithms, the simulations can be run in a semi-autonomous manner.

6.2 Parameters and Metrics

The parameters and metrics used in this exploration are a subset of those presented in Chapter 5 (Section 5.3), as shown in Tab. 6.1.

The range of parameters leads to a smaller parameter space to be explored and the simulations are only run for 10 seconds (simulation time), to enable rapid turnover of simulation iterations, towards an optimised value. This scenario produces 40 parameter sets that are explored through simulation.

The goal of the optimisation engine is to find a set of network parameters that remain within the constraints of packet reception success rate and end-to-end latency. The limits of these two values will obviously depend on the application, but for this work the limits are: packet reception success rate must remain higher than 75% and latency must remain below 90ms. From the results of Chapter 5 we know that the communication is related to the packet size, so we can adjust this to maintain higher QoS.

The parameters that can be varied by the device are transmission power, and therefore the transmission range, and packet production rate. Although these are only two values, the system could be configured to operate on all the available parameters in the MAC and PHY layers (as shown in Tab. 2.2 and Tab. 2.3).

6.3 Optimisation Algorithm

The use of optimisation algorithms to improve network operational efficiency has been widely used, as in the work of Song and Li [93], [94]. In their work, Song and Li have provided a theoretical framework for optimising the OFDM characteristics of wireless networks. The authors create a communication link between the MAC and PHY layers to transfer information gleaned from the sensing activities that take place in the network device.

The parallel method presented in Chapter 4 (page 69) was used to perform the simulations, such that the network settings represent a realistic IEEE 802.11p configuration. Given the number of simulations required for each iteration of the algorithm a small cluster (Mavrino, introduced on page 49) was used for the simulations.

The algorithm works by identifying each possible packet size for a parameter set and then finding the optimum packet size for that communication density. This is, in turn, fed back into the simulation where a new set of simulations is spawned that focus in on the optimum size. In this way the best packet size to communication density pairing can be found and set in the application.

The following pseudocode explains the algorithm further:

1. Define the problem (i.e. packet latency must be $< 90ms$)
2. Setup the master simulation with generic parameters
3. Launch a parallel exploration
4. Calculate results

For each simulation instance {

Is Problem OK (i.e. packet latency $< 90ms$)?

If yes, note communication density and individual parameters, else drop results

}

5. For all results where problem is OK calculate position of result on a graph where $(x, y) = (problem, communicationdensity)$

6. Analyse results to see which parameters are best fit for problem

7. Calculate new range of parameters to explore

GOTO 2. with reduced range of parameters

8. Is n_{th} run better than $n - 1_{th}$ run (in terms of problem definition)? If yes, GOTO 2. Else END.

6.4 Results

The results found were, as expected, that the greater the channel utilisation, the smaller the packet size was required to maintain a high QoS. Fig. 6.1 shows this for the full range of parameters studied.

Due to time constraints a further, more extensive, search of the parameter space was not available, but this shows how parameters can be optimised to produce higher QoS in VANET.

6.5 Conclusions

The application of this optimisation extends beyond lookup tables for network and application settings, and could be used to rapidly sense and adapt the network settings in a live system. As the technology attached to networking devices improves (in terms of computational power and memory resources) the complexity and number of parameters could increase to encompass the full range available.

This work intended to use a genetic algorithm and operate as a separate program that could run autonomously, but due to several constraints and problems with the NS-3 sub-system, it was not possible.

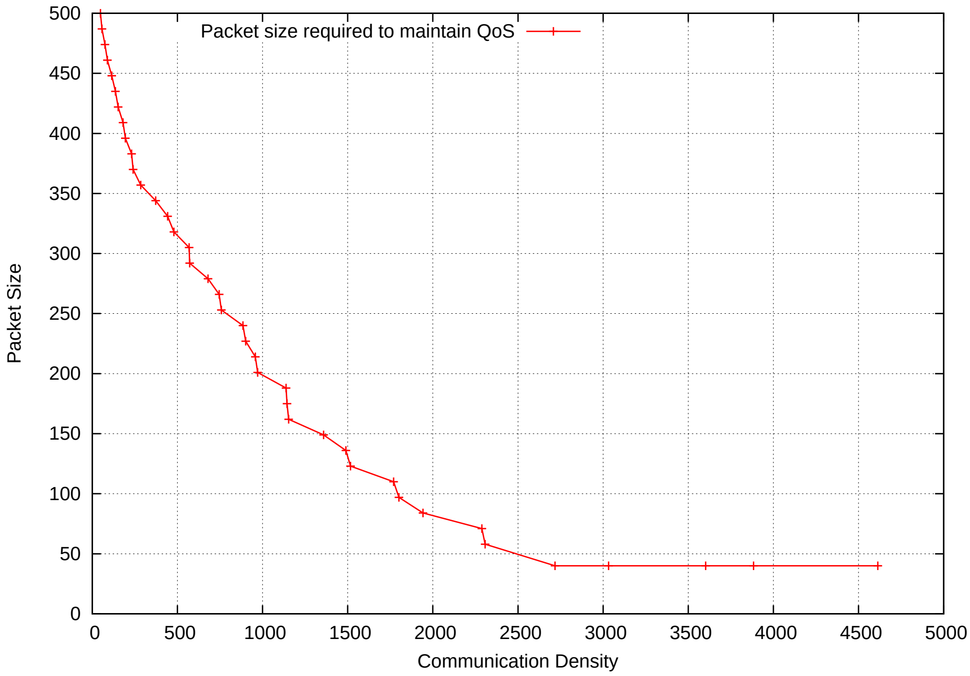


FIGURE 6.1: The figure shows that the packet size required to maintain latency below 90ms and packet reception rate above 75% drops as the communication density, and therefore channel utilisation, increases.

Chapter 7

Conclusions and Outlook

In this chapter the main findings of this thesis are summarised and the contributions made to the field of simulation for vehicular *ad hoc* networks are described, with discussion on the outlook for future work in this area. A critical evaluation of this work is presented and references to similar work in the field are shown to underline the contribution of this thesis.

7.1 Summary of the Thesis

The simulation and modelling of vehicular *ad hoc* networks requires a large amount of computational processing power, in order to perform accurate simulations of real-world scenarios. In this thesis high-performance computers have been extensively used, often with several thousand independent processing units, to achieve a deeper and more rigorous study of the available parameters and settings than previously explored in the pertinent literature.

The data and information in Chapter 2 (on page 19) presents the plethora of standards, specifications and studies already laid out for this particular area. Much of the work that has been done previous to this thesis looks at microscopic, one-to-one, interactions between vehicles and transmitting stations. In VANET, however, the interactions between thousands of vehicles and with millions of packets is more important to the applications running on a network. In this way the need for very large-scale and high-fidelity simulations for VANET research and development has been developed.

The congestion reduction algorithm developed in Chapter 3 (on page 52) shows how ITS applications using VANET communication can improve traffic movement and increase driver knowledge. The fact that it was possible to quickly and accurately deploy this algorithm in a tightly coupled simulation reinforces the capability of simulation over real-world experimentation. In order to fully test the algorithm it was also shown that the application stands up when increasing amounts of the simulated vehicles are not involved in the VANET.

The simulation system in Chapter 3 uses a lightweight network model that contains simplifications at the MAC layer. In order to produce high-fidelity simulations a fully functional model is required, and in Chapter 4 (page 69) the NS-3 network simulator was utilised, which contains complex and validated models for all of the layers in the OSI model. In NS-3 it was possible to present a deep study of the network parameters and variables by using a high-performance computer to run many simulations at once, using the algorithm designed for this purpose. The parameter search algorithm is built into the simulation program used and enables the parameter space to be searched without too much involvement from the user.

Chapters 5 and 6 (pages 80 and 105 respectively) show the advanced features of a parallel simulation methodology and provide accurate results for two particular scenarios. In Chapter 5 the simulation system that was developed showed that, across a massive parameter space, the individual parameters that make up a VANET scenario can be united and reduced to a single function. Chapter 6 shows an example of how this reduction can be used to optimise the network settings for increased quality of service, in terms of packet latency and successful reception rates.

7.2 Final Contributions

The main contributions of this thesis are listed in the Introduction (Chapter 1 on page 13) and so here the final output of this research is presented, with information regarding future use of the software that has been developed:

- - Enhanced lane change decision algorithm (from Chapter 3) based on the TrafficCom simulator (Treiber [10] and with telecommunications Nekovee and Bogason [42]) showing congestion reduction for obstructed highways.

- - Fully validated Two-ray ground propagation model for NS-3 (see A).

- - Proven collapse of transmission range, transmission rate and vehicular density for packet success rates and end-to-end latencies as a function of communication density, tested for a massive parameter space (Chapter 5).

- - A portable parallel algorithm for running many NS-3 simulations with different parameter sets to accomplish large scale simulation of VANET scenarios (developed in Chapter 4).

- - Evidence of how these contributions can be used to achieve optimisation of vehicular network settings, to enable improved network quality of service (see Chapter 6).

7.3 Critical Evaluation

The major focus of this thesis has been to perform high fidelity simulations of VANET in a short time for a range of scenarios. In specific situations, such as those shown in [95] where each simulation runs for 10 seconds, the fidelity or simulation size can be constrained to produce more results and collect more data. The scenarios examined in this thesis have been fitted with a specific balance of complexity and runtime, even when using high performance computing. This balance may not be suitable for all scenarios without specific fitting of the models and algorithms. A more generic approach would be to use just the core functionality of NS-3, but this wouldn't accurately model VANET for the scenarios shown in Chapters 3-6. Some work has been accomplished in the field of high performance simulation, such as Fujimoto et al. [96] and Mahajan et al. [97] which use highly fitted models for a specific range of scenarios. In contrast the work of Kosch et al. [98] uses a more generic approach, that applies to both highway and urban scenarios.

In this work the use of mobility, according to the Treiber IDM (page 30), has been setup for highway scenarios. As shown Treiber et al. [44] the empirical study of vehicular flow is mostly appropriate to highway scenarios and roads with very few junctions. As such, the work presented here is very pertinent to VANET applications that operate on highways and on trunk roads. The modular use of NS-3 does allow different mobility models to be incorporated into the VANET simulations covered in this thesis, to operate on urban scenarios.

The parameters that have been searched and analysed in this thesis reach a greater extent than any previous published work found. The number of parameters available to the simulation system will increase as network technology moves forward. The methodology developed to explore a large parameter space can be expanded to encompass future developments in this area.

7.4 Outlook

The field of VANET is still young compared with computer communication in general but the proliferation of devices, reduction in physical size and falling costs of wireless radios is enabling a new wave of small and ubiquitous devices, including vehicles. Future research in the field of VANET will use computer simulation to examine large and highly complex scenarios and applications. The work accomplished in this thesis has developed innovative software for high performance and large parameter search simulations coupled with novel applications and findings in this field.

The future use of the work presented in this thesis could follow the following paths:

- - Optimisation of VANET applications to achieve high and sustainable QoS using simulation results to tune the network across the full spectrum of variables and parameters in the protocol stack and in the physical and medium-access control layers.

- - Definition of new or improved MAC/PHY layer models in NS-3 and comparative studies through simulation to ensure successful operation in a large field of vehicles (exceeding thousands of vehicles).

Appendix A

Adding the Two-Ray Ground Reflection Model to NS3

The propagation of radiowaves is covered in Chapter 2 (on page 25). The NS-3 simulation system didn't have a model for the two-ray ground reflection model and so I contributed to the open-source software with a model written in C++. The model I contributed has been implemented in NS-3 since version 3.8 (released in June 2008).

The two-ray ground model extends the Friis free-space propagation model to incorporate the multi-path effects of reflection from the surface between transmitter and receiver.

The calculation of the two-ray ground reflection model measures the loss in signal strength as the wave moves through the medium. The received power is calculated as follows:

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \tag{A.1}$$

where: P_r is the received power, d is the distance from the transmitter, P_t is the transmitted power, G_t and G_r are the transmit and receive antenna gain, h_t and h_r are the transmit and receive antenna heights lambda (λ) is the wavelength of the transmitted signal and L is the system loss.

P_r, P_t, G_t, G_r are all measured in Watts for Eq. A.1 but in most simulations we wish to measure signal power in dBm (the power ratio in deciBels (dB) of the measured power referenced to 1 milliWatt (1mW)).

The conversion of Watts to dBm takes the form:

$$P = 10^{(x-30)/10} \tag{A.2}$$

and in the other direction:

$$x = 10log_{10}P + 30 \tag{A.3}$$

where P is the power in Watts and x is the power ratio in dBm.

A further extension from the Friis free-space model is the crossover distance calculation that measures the point where the signal will reflect off the ground, as shown in Fig. A.1.

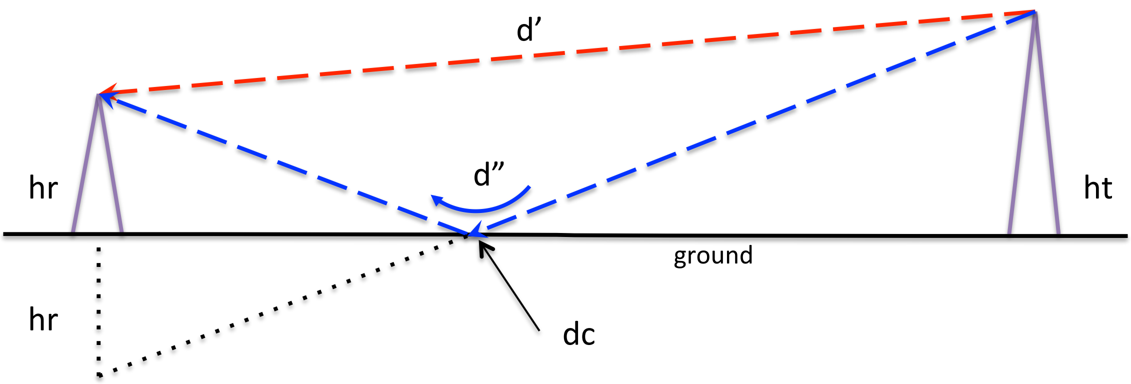


FIGURE A.1: The two-ray ground model takes the pathloss exponent from the path between transmitter and receiver where it has reflected of the ground (Earth's surface of other reflective material). The model calculates the reflection point as the distance where the path would reach the inverse height of the receiving tower.

The crossover distance d_c is calculated by:

$$d_c = \frac{4\pi h_r h_t}{\lambda} \tag{A.4}$$

If the measured distance between transmitter and receiver is less than the crossover distance the Friis free-space model is used. This is due to the oscillations that can be caused by the constructive and destructive action on the waves over short distances, as examined in Mizutani and Kohno [99].

The algorithm for calculating the pathloss and the received signal strength is as follows:

---

1. Take transmitted signal strength P_t
2. Calculate distance between sender and receiver d
3. Calculate the crossover distance dc

if $d > dc$ **then**

  4. Calculate two-ray pathloss l

else

  4. Calculate Friis pathloss l

end if

5. Calculate received signal strength $P_r = P_t - l$

The Doxygen documentation for the code can be found in [100].

The files I worked on are (as of NS-3.9):

./src/common/propagation-loss-model.h - Added header functions and inheritance classes

./src/common/propagation-loss-model.cc- Wrote main code for two-ray model with crossover distance calculation

./src/common/propagation-loss-model-test-suite.cc - Build some test cases for automated installation checking

Appendix B

Publications

In the work for Chapter 3 on congestion reduction the work was published in Hewer and Nekovee [6] for the 4th International Workshop on Vehicle-to-Vehicle Communications, Eindhoven, 2008. A substantially improved version was published and presented in Hewer and Nekovee [101] for the EuropeComm conference, London, 2009.

The addition of penetration rate analysis and extra data was published in Hewer et al. [79] in the EURASIP Journal on Advances in Signal Processing, 2010.

The work for chapter 4 was published in Hewer et al. [102] for the IEEE International Conference Advances in Computational Tools for Engineering Applications, Lebanon, 2009.

The work in Chapter 5 has been accepted [103] for IEEE Communications Letters (which ranks 15th in the top telecommunications journals, as measured by citation in 2004 [104]).

Appendix C

Parameters and Performance Metrics

This appendix shows the parameters and performance metrics used in this thesis, with a brief explanation of their particular meaning.

C.1 Parameters

C.1.1 Scenario Parameters

Road Length (m) - the length of the studied road/highway section, assumes a standard width of 3.5m.

Obstruction Location (m) - the position on the road where an obstruction occurs (for Chapter 3).

Number of Lanes - how many lanes the road contains and whether the direction of travel is the same or opposing.

Vehicle Velocity (m/s, mph, kph) - the speed at which vehicles are travelling. This can be an average across the field or per vehicle depending on the implementation.

Vehicular Density (vehicles per metre) - how many vehicles are present on a given section of road of a set length, where the density assumes uniform separation.

Number of Vehicles - the number of vehicles present in a simulated system regardless of any other parameter.

Vehicle Length (m) - the length of a vehicle in a simulation (for Chapter 3 this can vary between cars and trucks), usually set at 4m for cars.

Simulation Time Step (s) - how often the model updates itself and recalculates the next set of instructions.

Politeness Factor - a value that reduces the calculated advantage of a lane change or merge action to take place, which replicates the human factor of driving courtesy.

Penetration Rate (%) - the percentage of vehicles that have been equipped with radio equipment, which tests for realistic networks where only some new vehicles are radio equipped and therefore members of a VANET.

Simulation Time (s) - the amount of time that passes in a simulation.

Number of Processors/Cores - how many computer processors are used for the simulation; one processor may have 1-8 cores (currently).

Antenna Height (m) - the height at which the radio antenna is positioned, for use in the Two-Ray Ground Reflection propagation model, shown in Appendix. A.

C.1.2 Application Parameters

Transmission Method - a model that describes the way in which data is transmitted, according to either stochastic or probabilistic functions, or both.

Packet Transmission Rate (Hz) - the number of packets that are transmitted per second according to a preset distribution (in the case of Chapter 4-6 a Poisson distribution is used to represent unsynchronised transmission, keeping the same transmission rate).

Packet Size (b) - the size of the data packet being sent across through the network equipment, in most cases this represents the payload and not the added data from encapsulation at the layers of the OSI model.

C.1.3 MAC Layer Parameters

MAC Model - the particular model that represents the MAC layer, with specific functions for each of the IEEE 802.11 standards.

CTS Timeout (ms) - as specified in the particular IEEE 802.11 standard, the CTS timeout represents the time to wait until a CTS authorisation is received from a possible recipient, as shown in Table 2.2 on page 45.

ACK Timeout (ms) - acknowledgment messages are sent to advise of a successful packet reception, in *ad hoc* and infrastructure-based networks, where the timeout is the time a sending station waits before assuming a packet has been lost in transmission.

SIFS (ms) - the Short Inter-Frame Spacing time refers to the minimum interval between receiving a data and sending an acknowledgement, to mitigate fluctuations on the medium due to interference etc..

DIFS (ms) - the Distribution (coordination function) Inter-Frame Space is the time allowed for DCF activity in the MAC layer to sense the medium and determine it's state.

Slot Time (ms) - the slot time is twice the theoretical time taken to send a single pulse between sender and receiver, and enable the MAC layer to calculate how long it will require the medium for.

C.1.4 PHY Layer Parameters

Transmission Range (m) - a pseudo-realistic distance within which a transmitted packet should be received. Inaccurate due to the complex calculation of packet reception according to propagation pathless and interference or noise on the medium, but useful for defining a constraint on calculating packet success in a simulated network.

Transmission Power (dBm, W) - the equivalent isotropically emitted power (EIRP) of a radio signal from a transmitter's antenna measured in decibel metres (dBm) or Watts (W).

Channel Bandwidth (Mbps, kbps) - given the coding mechanism to put data onto the medium, the channel bandwidth is the maximum capacity of the medium, i.e. the amount of data the medium can hold per second, measured in kilobits per second or megabits per second.

Centre Frequency (MHz, GHz) - the central frequency represents the centre of the used channel in a frequency range, i.e. the 5.9GHz frequency specified for DSRC ranges from 5.860 to 5.920GHz in 10MHz channels (real range is therefore 5.850 to 5.925GHz).

Wavelength (mm) - the wavelength is the distance between the period of a single waveform (the distance between two peaks in a sinusoidal wave) and is calculated by $\lambda = \frac{C}{f}$ where C is the speed of light and f is the centre frequency.

Transmission Method - the exact way in which binary data is encoded onto the medium varies across implementation using, for example, phase-shift keying and amplitude modulation. More detail can be found in Katariya et al. [105].

Propagation Model - the propagation pathless model calculate the loss of energy of a radio wave as it travels away from the transmitter, further explained in Chapter 2.1.3 on page 25.

Error-Rate Model - the error rate model calculates the success of any packet reception through a stochastic (dependent on the medium SNIR/Noise level) and a probabilistic method, to accurately represent the chance of loss on a utilised medium.

Energy Detection Threshold (dBm, W) - the energy detection threshold is the power level sensed on the medium that enables the PHY layer to switch from IDLE to CCA_BUSY (full details of the PHY states are in Section 2.5.1 on page 47). This value is device-dependent and varies according to the sensitivity of the network device.

CCA1 Threshold (dBm, W) - the CCA1 threshold is often higher than the EDT and is the threshold above which power on the medium enables the PHY to switch from IDLE or CCA_BUSY to the SYNC state, in order to begin reception of data. As with the EDT this value is device-dependent and is often defined by the manufacturer.

C.2 Performance Metrics

Average Velocity (m/s, mph, kph) - the average velocity is a mean calculation of all the velocities in a simulation, which gives an overall perspective on the flow of vehicles along a highway.

Vehicle Flow Profile - as shown in Fig. 3.2 on page 59 the flow profile is the percentage of $\frac{CarsEntering}{CarsLeaving}$. This metric shows how many vehicles are being held in the simulated system, due to congestion, and also shows the percentage of cars that have successfully traversed the obstruction around which congestion develops.

Vehicular Throughput - vehicular throughput is a more commonly used metric than a vehicle flow profile; it represents the actual throughput of vehicles at a given position in the simulated system.

Position of Lane Change/Merge (m) - this is the position where a vehicle makes a lane-change/merge action and is indicative of when it receives information regarding an obstruction, or just if there is an advantage to change lanes at that position.

Packet End-to-End Latency (ms, ns) - the packet end-to-end latency is a measurement of a number of actions. In this thesis the end-to-end latency of a packet is the time from entering the transmitting stations MAC layer to the packet leaving the receiving stations PHY layer, thereby encompassing the DCF function, the physical encoding and transmission on the medium and the time taken to receive data off the medium by the receiver. The definition of end-to-end latency varies through the literature and so should be well-studied if using for comparison.

Average Latency (ms, ns) - the average latency is a mean calculation of the individual end-to-end latencies gathered in a simulation. In this thesis only successfully received packets provide an end-to-end value as any dropped packets could supply an incomplete traversal of the actions that are measured, skewing the average.

Packet Success Rate (%) - the packet success rate is a percentage of packets sent and packets successfully received, as reported by the transmitter and receiver. This rate takes into account packets that are lost through collision on the medium and errors in reception and is not related to packets that are not receivable (i.e. outside the range of detection).

Communication Density - the communication density, as explained in Section 5.3.2.3 on page 90, is calculated at the beginning of the simulation from the specified system parameters (road length, transmission range, transmission rate and vehicular density). This metric represents the number of transmissions that are made on a section of highway per second.

Number of Neighbours - the number of neighbours that any vehicle will have depends upon a number of factors, primarily which vehicles are within the detection range (Fig. 2.9 on page 46) of a transmitting station. The propagation model that is being used, coupled with the error-rate model will give a realistic distance below which vehicles can be classed as neighbours. When a receiver is below this range it does not guarantee successful reception of the packet, but gives an accurate value for expected reception.

Bibliography

[1] IEEE. IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pages C1–1184, 2007. doi: 10.1109/IEEESTD.2007.373646.

[2] IETF. The Internet Engineering Task Force - MANET Working Group, Available: http://www.ietf.org/html.charters/manet-charter.html. *IETF*, 2005.

[3] Manfred Broy, Ingolf H. Kruger, Alexander Pretschner, and Christian Salzmann. Engineering Automotive Software. *Proceedings of the IEEE*, 95(2):356–373, Feb. 2007. ISSN 0018-9219. doi: 10.1109/JPROC.2006.888386.

[4] Darren Waters. Connected cars 'promise safer roads'. *BBC News 2007*, 2007. URL http://news.bbc.co.uk/1/hi/technology/6274974.stm.

[5] Marc Torrent-Moreno. Inter Vehicle Communications, Achieving Safety in a Distributed Wireless Environment: Challenges, Systems and Protocols. In *Ph.D. Thesis at Universität Karlsruhe*, 2007.

[6] Thomas D. Hewer and Maziar Nekovee. Congestion Reduction Using Ad hoc Message Dissemination in Vehicular Networks. In *4th International Workshop on Vehicle-to-Vehicle Communications*, pages 1–7, 2008. URL http://arxiv.org/abs/0807.2829.

[7] Andrew Chen, Behrooz Khorashadi, Chen-Nee Chuah, Dipak Ghosal, and Michael Zhang. Smoothing Vehicular Traffic Flow Using Vehicular-Based Ad Hoc Networking & Computing Grid (VGrid). *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 349–354, 2006. doi: 10.1109/ITSC.2006.1706766.

[8] nVidia. The Third Pillar of Science. *Youtube video website*, 2010. URL www.youtube.com/watch?v=3DbAB2ChDBw.

[9] John Heidemann, Kevin Mills, and Sri Kumar. Expanding confidence in network simulations. *Network, IEEE*, 15(5):58–63, Sep/Oct 2001. ISSN 0890-8044. doi: 10.1109/65.953234.

[10] Martin Treiber. Microsimulation of road traffic, View at: http://www.traffic-simulation.de/. *traffic-simulation.de*, 2010.

[11] Arne Kesting, Martin Treiber, and Dirk Helbing. General Lane-Changing Model MOBIL for Car-Following Models. *Traffic Flow Theory 2007*, pages pp 86–94, 1999. doi: http://dx.doi.org/10.3141/1999-10.

[12] Daniel Jiang, Qi Chen, and Luca Delgrossi. Communication Density: A Channel Load Metric for Vehicular Communications Research. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*, pages 1 –8, oct. 2007. doi: 10.1109/MOBHOC.2007.4428734.

[13] Chris Davies. Will future car manufacturing incorporate Wi-Fi? *Manufacturing Digital News*, March 2011. URL http://www.manufacturingdigital.com/tags/saab/will-future-car-manufacturing-incorporate-wi-fi.

[14] IEEE. IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Amendment 5: Media Access Control Parameters, Physical Layers, and Management Parameters for Energy-Efficient Ethernet. *IEEE Std 802.3az-2010 (Amendment to IEEE Std 802.3-2008)*, pages 1 –302, 27 2010. doi: 10.1109/IEEESTD.2010.5621025.

[15] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[16] Matthew Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly Media, 2005.

[17] Michael Donahoo and Kenneth Calvert. *TCP/IP Sockets in C Practical Guide for Programmers*. Morgan Kaufmann, San Mateo, CA, 2002.

[18] David B. Johnson, David A. Maltz, and Yi-Chun Hu. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). Technical report, IETF MANET Working Group, 2004.

[19] Keighley Instruments Inc. An Introduction to Orthogonal Frequency Division Multiplex Technology. *IEEE Online http://www.ieee.li/pdf/viewgraphs/*, 2004.

[20] Greg Goth. Delay-Tolerant Network Technologies Coming Together. *Distributed Systems Online, IEEE*, 7(8):2–2, Aug. 2006. ISSN 1541-4922. doi: 10.1109/MDSO.2006.49.

[21] Maziar Nekovee. Modeling the Spread of Worm Epidemics in Vehicular Ad Hoc Networks. *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, 2:841–845, 7-10 May 2006. ISSN 1550-2252. doi: 10.1109/VETECS.2006.1682943.

[22] Audi, BMW, DaimlerChrysler, Fiat, Renault, and VW. Car 2 Car Communication Consortium, available: http://www.car-to-car.org/. 2008.

[23] Christian Lochert, Björn Scheuermann, Christian Wewetzer, Andreas Luebke, and Martin Mauve. Data aggregation and roadside unit placement for a vanet traffic information system. In *VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, pages 58–65, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-191-0. doi: http://doi.acm.org/10.1145/1410043.1410054.

[24] Marina Aguado, Jon Matias, Eduardo Jacob, and Marion Berbineau. The WiMAX ASN Network in the V2I Scenario. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1 –5, sept. 2008. doi: 10.1109/VETECF.2008.457.

[25] IEEE. Ieee standard for local and metropolitan area networks part 16: Air interface for broadband wireless access systems. *IEEE Std 802.16-2009 (Revision of IEEE Std 802.16-2004)*, pages C1 –2004, 29 2009. doi: 10.1109/IEEESTD.2009.5062485.

[26] Harald T. Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254 – 256, May 1946. ISSN 0096-8390. doi: 10.1109/JRPROC.1946.234568.

[27] David C. Hogg. Fun with the Friis free-space transmission formula. *Antennas and Propagation Magazine, IEEE*, 35(4):33 –35, aug. 1993. ISSN 1045-9243. doi: 10.1109/74.229847.

[28] S. A. Hosseini Tabatabaei, M. Fleury, N. N. Qadri, and M. Ghanbari. Improving propagation modeling in urban environments for vehicular ad hoc networks. *Intelligent Transportation Systems, IEEE Transactions on*, PP(99):1 –12, 2011. ISSN 1524-9050. doi: 10.1109/TITS.2011.2143707.

[29] Garrey Learmonth and JoAnne Holliday. MCCA: Multihop Clear Channel Assessment In 802.11 DCF. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1 –2, jan. 2010. doi: 10.1109/CCNC.2010.5421699.

[30] Yahong Rosa Zheng and Chengshan Xiao. Simulation models with correct statistical properties for Rayleigh fading channels. *Communications, IEEE Transactions on*, 51(6):920–928, June 2003. ISSN 0090-6778. doi: 10.1109/TCOMM.2003.813259.

[31] Yacine Khaled, Bertrand Ducourthial, and Mohamed Shawky. IEEE 802.11 performances for inter-vehicle communication networks. *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, 5:2925–2929 Vol. 5, 30 May-1 June 2005. ISSN 1550-2252. doi: 10.1109/VETECS.2005.1543882.

[32] Daniel Jiang, Vikas Taliwal, Andreas Meier, Wieland Holfelder, and Ralf Hertwich. Design of 5.9 ghz dsrc-based vehicular safety communication. *Wireless Communications, IEEE*, 13(5):36 –43, October 2006. ISSN 1536-1284. doi: 10.1109/WC-M.2006.250356.

[33] Vikas Taliwal, Daniel Jiang, Heiko Mangold, Chi Chen, and Raja Sengupta. Empirical determination of channel characteristics for DSRC vehicle-to-vehicle communication. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 88–88, New York, NY, USA, 2004. ACM. ISBN 1-58113-922-5. doi: http://doi.acm.org/10.1145/1023875.1023890.

[34] CEN. Dedicated short-range communication. Physical layer using microwave at 5.8 GHz. *EN 12253:2004*, 2004.

[35] CEN. Dedicated Short-Range Communication (DSRC) - DSRC Data link layer: Medium Access and Logical Link Control. *EN 12795:2002*, 2002.

[36] CEN. Dedicated Short-Range Communication - Application layer. *EN 12834:2002*, 2002.

[37] Jijun Yin, Tamer ElBatt, Gavin Yeung, Bo Ryu, Stephen Habermas, Hariharan Krishn, and Timothy Talty. Performance Evaluation of Safety Applications over DSRC Vehicular Ad Hoc Networks. *VANET '04*, 2003.

[38] Stefan Eichler. Performance Evaluation of the IEEE 802.11p WAVE Communication Standard. *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 2199–2203, 30 2007-Oct. 3 2007. ISSN 1090-3038. doi: 10.1109/VETECF.2007.461.

[39] Youwei Zhang, Ian L. Tan, Carl Chun, Ken Laberteaux, and Ahmad Bahai. A differential OFDM approach to coherence time mitigation in DSRC. In *VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, pages 1–6, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-191-0. doi: http://doi.acm.org/10.1145/1410043.1410045.

[40] Katrin Bilstrup, Elisabeth Uhlemann, Erik G. Strom, and Urban Bilstrup. Evaluation of the IEEE 802.11p MAC Method for Vehicle-to-Vehicle Communication. *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5, Sept. 2008. ISSN 1090-3038. doi: 10.1109/VETECF.2008.446.

[41] Panos Papadimitratos, Levente Buttyan, Jean-Pierre Hubaux, Frank Kargl, Antonio Kung, and Maxim Raya. Architecture for Secure and Private Vehicular Communications. *Telecommunications, 2007. ITST '07. 7th International Conference on ITS*, pages 1–6, June 2007. doi: 10.1109/ITST.2007.4295890.

[42] Maziar Nekovee and Benedict Bogason. Reliable and Effcient Information Dissemination in Intermittently Connected Vehicular Adhoc Networks. *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pages 2486–2490, 22-25 April 2007. ISSN 1550-2252. doi: 10.1109/VETECS.2007.512.

[43] NASA. TRANSIMS (TRansportation ANalysis SIMulation System) Available:http://www.transims-opensource.net/. *TRANSIMS OPEN SOURCE PROJECT SITE*, 2007.

[44] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E*,

62:1805, 2000. URL http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0002177.

[45] Chiyomi Miyajima, Yoshihiro Nishiwaki, Koji Ozawa, Toshihiro Wakita, Katsunobu Itou, Kazuya Takeda, and Fumitada Itakura. Driver Modeling Based on Driving Behavior and Its Evaluation in Driver Identification. *Proceedings of the IEEE*, 95(2):427–437, Feb. 2007. ISSN 0018-9219. doi: 10.1109/JPROC.2006.888405.

[46] Min Young Cho, A.J. Lichtenberg, and M.A. Lieberman. Minimum stopping distance for linear control of an automatic car-following system. *Vehicular Technology, IEEE Transactions on*, 45(2):383–390, May 1996. ISSN 0018-9545. doi: 10.1109/25.492913.

[47] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii. Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications. *Intelligent Transportation Systems, IEEE Transactions on*, 3(3):155 – 161, sep 2002. ISSN 1524-9050. doi: 10.1109/TITS.2002.802929.

[48] A. Kanaris, E.B. Kosmatopoulos, and P.A. Loannou. Strategies and spacing requirements for lane changing and merging in automated highway systems. *Vehicular Technology, IEEE Transactions on*, 50(6):1568 –1581, nov 2001. ISSN 0018-9545. doi: 10.1109/25.966586.

[49] Daniel Krajzewicz, Georg Hertkorn, C. Rössel, and Peter Wagner. Simulation of urban mobility (sumo) available at: http://sourceforge.net/projects/sumo. *SUMO*, 2002.

[50] Daniel Krajzewicz, Georg Hertkorn, C. Rössel, and Peter Wagner. SUMO (Simulation of Urban MObility) - An Open-Source Traffic Simulation. *Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM20002)*, pages pp. 183–187, 2002.

[51] Michal Piorkowski, Maxim Raya, Ada Lugo, Panos Papadimitratos, Matthias Grossglauser, and Jean-Pierre Hubaux. TraNS: Realistic Joint Traffic and Network Simulator for VANETs. *ACM SIGMOBILE Mobile Computing and Communications Review*, 2005. doi: NA.

[52] Qi Chen, Felix Schmidt-Eisenlohr, Daniel Jiang, Marc Torrent-Moreno, Luca Delgrossi, and Hannes Hartenstein. Overhaul of IEEE 802.11 modeling and simulation in ns-2. In *MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, pages 159–168, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-851-0. doi: http://doi.acm.org/10.1145/1298126.1298155.

[53] Thomas D. Hewer, Maziar Nekovee, Radhika S. Saksena, and Peter V. Coveney. Challenges of Large-scale Vehicular and Mobile Ad hoc Network Simulation. In *UK E-Science All Hands Meeting*, 2008.

[54] Kevin Fall and Kannan Varadhan. *The ns Manual (formerly ns Notes and Documentation)*. VINT, 2006.

[55] Nils Aschenbruck et al. BonnMotion a Mobility Scenario Generation and Analysis Tool. Technical report, University of Bonn, 2009.

[56] Roger Krügler and Holger Fūβler. HWGui - Visualizing, Evaluating and Transforming Movement Patterns of Vehicles on Highways. Technical report, University of Mannheim, 2008.

[57] Jerome Haerri, Fethi Filali, and Christian Bonnet. Mobility models for vehicular ad hoc networks: a survey and taxonomy. *Accepted for IEEE Communications Surveys and Tutorials (epublication)*, 2008.

[58] Mathieu Lacage and Thomas R. Henderson et al. The Network Simulator - ns-2, Available: http://www.isi.edu/nsnam/ns. *NS-2*, 1998.

[59] Daniel Jiang and Luca Delgrossi. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2036–2040, May 2008. ISSN 1550-2252. doi: 10.1109/VETECS.2008.458.

[60] Richard Bossom, Roberto Brignolo, Thierry Ernst, Knut Evensen, Alexander Frötscher, Wolfgang Höfs, Juhani Jääskeläinen, Zeljko Jeftic, Paul Kompfner, Timo Kosch, Ilse Kulp, Antonio Kung, Abdel-Kader Mokaddem, Andreas Schalk, Elisabeth Uhlemann, and Christian Wewetzer. D31 European ITS Communication Architecture. Technical report, Information Society Technologies Communication for e-Safety, 2009.

[61] Marc Torrent-Moreno, Paolo Santi, and Hannes Hartenstein. Distributed Fair Transmit Power Adjustment for Vehicular Ad Hoc Networks. *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, 2:479–488, Sept. 2006. doi: 10.1109/SAHCN.2006.288504.

[62] Andras Varga. Using the OMNeT++ discrete event simulation system in education. *Education, IEEE Transactions on*, 42(4):11 pp.–, Nov 1999. ISSN 0018-9359. doi: 10.1109/13.804564.

[63] K. Kucuk, A. Kavak, and H. Yigit. A Smart Antenna Module Using OMNeT++ for Wireless Sensor Network Simulation. *Wireless Communication Systems, 2007. ISWCS 2007. 4th International Symposium on*, pages 747–751, Oct. 2007. doi: 10.1109/ISWCS.2007.4392440.

[64] Rimon Barr, Zygmunt J. Haas, and Robbert van Renesse. JiST: an efficient approach to simulation using virtual machines: Research Articles. *Softw. Pract. Exper.*, 35(6):539–576, 2005. ISSN 0038-0644. doi: http://dx.doi.org/10.1002/spe.v35:6.

[65] Trishla Sutaria, Imad Mahgoub, Ali Humos, and Ahmed Badi. Implementation of an Energy Model for JiST/SWANS Wireless Network Simulator. *Networking, 2007. ICN '07. Sixth International Conference on*, pages 24–24, 22-28 April 2007. doi: 10.1109/ICN.2007.47.

[66] Elmar Schoch. VANET Simulations with JiST/SWANS. In *SEVECOM Kick-off workshop*, 2006.

[67] Jorge Nuevo. A Comprehensible GloMoSim Tutorial. Technical report, INRS - Universite du Quebec, 2004.

[68] Rajive Bagrodia, Richard Meyer, Mineo Takai, Yu-An Chen, Xiang Zeng, Jay Martin, and Ha Yoon Song. Parsec: a parallel simulation environment for complex systems. *Computer*, 31(10):77–85, Oct 1998. ISSN 0018-9162. doi: 10.1109/2.722293.

[69] Rajive Bagrodia. QualNet simulator, 2001.

[70] Jost Weinmiller, Hagen Woesner, and Adam Wolisz. Analyzing and Improving the IEEE 802.11-MAC Protocol for Wireless LANs. In *Proceedings of the 4th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '96)*, pages 200–206, 1996. URL citeseer.ist.psu.edu/weinmiller96analyzing.html.

[71] IEEE. IEEE Draft Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment : Wireless Access in Vehicular Environments. *IEEE Unapproved Draft Std P802.11p /D10.0, Jan 2010*, 2010.

[72] IEEE. IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements. *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)*, pages 1 –189, 2005. doi: 10.1109/IEEESTD.2005.97890.

[73] Mathieu Lacage and Thomas R. Henderson. Yet another network simulator. In *WNS2 '06: Proceeding from the 2006 workshop on ns-2: the IP network simulator*, page 12, New York, NY, USA, 2006. ACM. ISBN 1-59593-508-8. doi: http://doi.acm.org/10.1145/1190455.1190467.

[74] NS-3 Developers. Interference Helper Source Code from NS-3. *NS-3 Doxygen Documentation*, 2009. URL http://www.nsnam.org/doxygen-release/classns3_1_1_interference_helper.html.

[75] Jose Miguel, Agustin Arruabarrena, Ramon Beivide, and Jose A. B. Fortes. An empirical evaluation of techniques for parallel discrete-event simulation of interconnnection networks. *Parallel and Distributed Processing, 1996. PDP '96. Proceedings of the Fourth Euromicro Workshop on*, pages 219–226, 24-26 Jan 1996. doi: 10.1109/EMPDP.1996.500590.

[76] NS-3 Developers. NS-3.8 Features. *NS-3 Blog Website*, 2010. URL http://nsnam.blogspot.com/2010/05/ns-38-released.html.

[77] Edgar Gabriel, Graham E. Fagg, George Bosilca, Thara Angskun, Jack J. Dongarra, Jeffrey M. Squyres, Vishal Sahay, Prabhanjan Kambadur, Brian Barrett, Andrew Lumsdaine, Ralph H. Castain, David J. Daniel, Richard L. Graham, and Timothy S. Woodall. Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation. In *Proceedings, 11th European PVM/MPI Users' Group Meeting*, pages 97–104, Budapest, Hungary, September 2004.

[78] David Millward. Number of cars declines for the first time since Second World War. *The Telegraph Newspaper*, April 2010. URL http://www.telegraph.co.uk/motoring/news/7563297/Number-of-cars-declines-for-the-first-time-since-Second-World-War.html.

[79] Thomas D. Hewer, Maziar Nekovee, and Peter V. Coveney. Reducing Congestion in Obstructed Highways with Traffic Data Dissemination Using Ad hoc Vehicular Networks. *EURASIP Journal on Advances in Signal Processing*, 2010:10, July 2010. doi: 10.1155/2010/169503. URL http://www.hindawi.com/journals/asp/2010/169503.html.

[80] Monika Büscher, Paul Coulton, Christos Efstratiou, Hans Gellersen, Drew Hemment, Rashid Mehmood, and Daniela Sangiorgi. Intelligent Mobility Systems: Some Socio-technical Challenges and Opportunities. In R. Mehmood, E. Cerqueira, R. Piesiewicz, & I. Chlamtac, editor, *Communications Infrastructure. Systems and Applications in Europe*, pages 140–152, 2009. doi: 10.1007/978-3-642-11284-3_15.

[81] Mathieu Lacage and Thomas R. Henderson et al. The Network Simulator 3 (NS-3), Available: http://www.nsnam.org. *NS-3*, 2008.

[82] Jeremy J. Blum, Azim Eskandarian, and Lance J. Hoffman. Challenges of intervehicle ad hoc networks. *Intelligent Transportation Systems, IEEE Transactions on*, 5(4):347–351, Dec. 2004. ISSN 1524-9050. doi: 10.1109/TITS.2004.838218.

[83] Maziar Nekovee. Epidemic algorithms for reliable and efficient information dissemination in vehicular ad hoc networks. *Intelligent Transport Systems, IET*, 3 (2):104 –110, June 2009. ISSN 1751-956X. doi: 10.1049/iet-its:20070061.

[84] Thomas D.C. Little and Ashish Agarwal. An information propagation scheme for VANETs. *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 155–160, 13-15 Sept. 2005. doi: 10.1109/ITSC.2005.1520039.

[85] ETSI. ETSI TS 102 486-1-2: Intelligent Transport Systems (ITS); Road Transport and Traffic Telematics (RTTT); Test specifications for Dedicated Short Range Communication (DSRC) transmission equipment; Part 1: DSRC data link layer: medium access and logical link control; Sub-Part 2: Test Suite Structure and Test Purposes (TSSTP). *ETSI Technical Committee Intelligent Transport System(ITS)*, 2008.

[86] EU. Cars That Talk: Commission earmarks single radio frequency for road safety and traffic management. *Europa: The Official Website of the EU*, August 2008. URL http://europa.eu/rapid/pressReleasesAction.do?reference=IP/08/1240&format=HTML&aged=0&language=EN&guiLanguage=en.

[87] IEEE. IEEE 1609 WAVE Standard. *US Dept. of Transport*, 2009. URL http://www.standards.its.dot.gov/fact_sheet.asp?f=80.

[88] Subir Biswas, Raymond Tatchikou, and Francois Dion. Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety. *Communications Magazine, IEEE*, 44(1):74–82, Jan. 2006. ISSN 0163-6804. doi: 10.1109/MCOM.2006.1580935.

[89] Marc Torrent-Moreno. Inter-vehicle communications: assessing information dissemination under safety constraints. In *Wireless on Demand Network Systems and Services, 2007. WONS '07. Fourth Annual Conference on, Proceedings of,*, pages 59 –64, Jan. 2007. doi: 10.1109/WONS.2007.340471.

[90] Moritz Killat and Hannes Hartenstein. An empirical model for probability of packet reception in vehicular ad hoc networks. *EURASIP J. Wirel. Commun. Netw.*, 2009:1–12, 2009. ISSN 1687-1472. doi: http://dx.doi.org/10.1155/2009/721301.

[91] Mate Boban, Ozan K. Tonguz, and Joao Barros. Unicast Communication in Vehicular ad hoc Networks: A Reality Check. *Communications Letters, IEEE*, 13 (12):995 –997, dec. 2009. ISSN 1089-7798. doi: 10.1109/LCOMM.2009.12.091497.

[92] Endroyono Hendrantoro and Gamantyo Hendrantoro. Cross-layer optimization performance evaluation of OFDM broadband network on millimeter wave channels. In *5th IFIP International Conference on Wireless and Optical Communications Networks, 2008. WOCN '08.*, pages 1–5, May 2008. doi: 10.1109/WOCN.2008.4542512.

[93] Guocong Song and Ye Li. Cross-layer optimization for OFDM wireless networks-part I: theoretical framework. *Wireless Communications, IEEE Transactions on*, 4(2):614–624, March 2005. ISSN 1536-1276. doi: 10.1109/TWC.2004.843065.

[94] Guocong Song and Ye Li. Cross-layer optimization for OFDM wireless networks-part II: algorithm development. *Wireless Communications, IEEE Transactions on*, 4(2):625–634, March 2005. ISSN 1536-1276. doi: 10.1109/TWC.2004.843067.

[95] Qi Chen, Daniel Jiang, Vikas Taliwal, and Luca Delgrossi. IEEE 802.11 based vehicular communication simulation design for NS-2. In *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 50–56, New York, NY, USA, 2006. ACM. ISBN 1-59593-540-1. doi: http://doi.acm.org/10.1145/1161064.1161073.

[96] Richard M. Fujimoto, Kalyan Perumalla, Alfred Park, Hao Wu, Mostafa H. Ammar, and George F. Riley. Large-scale network simulation: how big? how fast? *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on*, pages 116–123, Oct. 2003. ISSN 1526-7539. doi: 10.1109/MASCOT.2003.1240649.

[97] Atulya Mahajan, Niranjan Potnis, Kartik Gopalan, and Andy Wang. Modeling vanet deployment in urban settings. In *MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, pages 151–158, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-851-0. doi: http://doi.acm.org/10.1145/1298126.1298154.

[98] Timo Kosch, Christian J. Adler, Stephan Eichler, Christoph Schroth, and Markus Strassberger. The scalability problem of vehicular ad hoc networks and how to solve it. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 13(5):22–28, October 2006. ISSN 1536-1284. doi: 10.1109/WC-M.2006.250354.

[99] Katsuya Mizutani and Ryuji Kohno. Analysis of Multipath Fading Due to Two-Ray Fading and Vertical Fluctuation of the Vehicles in ITS Inter-Vehicle Communications. In *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*, pages 318 – 323, 2002. doi: 10.1109/ITSC.2002.1041236.

[100] Thomas D. Hewer. Two-Ray Ground Reflection Propagation Loss Model. *NS-3 Doxygen Documentation*, 2009. URL http://www.nsnam.org/doxygen-release/classns3_1_1_two_ray_ground_propagation_loss_model.html.

[101] Thomas D. Hewer and Maziar Nekovee. Congestion Control and Reduction Using Ad Hoc V2V Traffic State Dissemination. In *EuropeComm*, 2009.

[102] Thomas D. Hewer, Maziar Nekovee, and Peter V. Coveney. Parameter exploration in parallel for dynamic vehicular network efficiency. In *Advances in Computational Tools for Engineering Applications, 2009. ACTEA '09. International Conference on*, pages 16–20, July 2009. doi: 10.1109/ACTEA.2009.5227879.

[103] T. Hewer, M. Nekovee, and P. Coveney. Universality of Performance Parameters in Vehicular *ad hoc* Networks. *Communications Letters, IEEE*, PP(99):1 –3, 2011. ISSN 1089-7798. doi: 10.1109/LCOMM.2011.062911.110710.

[104] IEEE. IEEE Communications Letters, Available: http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=4234. *IEEE Communications Letters*, 2011.

[105] Abhishek Katariya, Amita Yadav, and Neha Jain. Performance Elevation Criteria for OFDM under AWGN Fading Channel using IEEE 802.11a. *International Journal of Soft Computing and Engineering (IJSCE)*, 1(3):10–13, July 2011. ISSN 2231-2307.